LİSELER İÇİN
YAPAY ZEKÂ

NECMİ GÜRSAKAL
&
ChatGPT

Bursa Erkek Liselilere

İÇİNDEKİLER

GİRİŞ

Merhaba Yapay Zekâ
Bu sistemler sizi sizden daha iyi nasıl tanıyor?
Yapay Zekâ Nedir, Ne Değildir?
Zekâ Nedir? İnsan ve Makine
Günlük Hayatta Yapay Zekâ
Yapay Zekâ Efsaneleri ve Gerçekler
Dar Yapay Zekâ ve Genel Yapay Zekâ
Yapay Zekâ ile Dene
Bölüm Özeti
Soru-Cevap

MAKİNE NASIL ÖĞRENİR? (TEMEL KAVRAMLAR)

Veri: Yeni Dünyanın Yakıtı
Veri Nedir? Türleri
Veri Toplama, Temizleme ve Etiketleme
Büyük Veri Kavramı
Veri Kalitesi ve Önyargı
Basit Veri Örnekleri ve Mini Uygulamalar
Öğrenme Yöntemleri
Denetimli Öğrenme
Denetimsiz Öğrenme
Pekiştirmeli Öğrenme
Algoritma Nedir?
Makine Öğrenmesinin Günlük Hayat Örnekleri
Bölüm Özeti
Soru-Cevap

GÖREN VE KONUŞAN MAKİNELER

Bilgisayarlı Görü (Computer Vision)
Doğal Dil İşleme (Natural Language Processing – NLP)
Bölüm Özeti
Soru-Cevap

YAPAY ZEKÂ NERELERDE KULLANILIR?

Sağlık: İnsan Hayatını Kurtaran Algoritmalar
Ulaşım ve Otonom Sistemler: Geleceğin Trafiği
Oyunlar: Yapay Zekâ ile Dinamik Deneyimler
Sanat ve Yaratıcılık: İnsan Hayal Gücünün Ortağı
Eğitim: Kişiselleştirilmiş Öğrenme ve Analiz
Yapay Zekânın Çapraz Kullanım Örnekleri
Gelecek Perspektifi
Bölüm Özeti
Soru Cevap

YAPAY ZEKÂ NASIL ÖĞRENİR?

Eğitim ve Test Verisi
Hata ve Öğrenme İlişkisi
Basit Bir Modelin Adım Adım Kurulması
Aşırı Öğrenme (Overfitting) Fikri
Özyinelemeli Öğrenme Nedir?
Yapay Zekâ Neden Tekrar Tekrar Dener?
İlk Tahmin Neden Hatalı Olur?
Hata Nasıl Ölçülür?
Model Kendini Nasıl Düzeltir?
Aynı Veriyi Tekrar Tekrar Kullanmak Neden İşe Yarar?
“Öğrenme = Düzeltme Süreci” Fikri
Örnek Kod ile Sezgisel Gösterim
Hata ve Öğrenmenin Hayatımızdaki Karşılığı
Bölüm Özeti
Soru Cevap

YAPAY ZEKÂ İLE YAŞAMAK (OKURYAZARLIK)

Yapay Zekâyı Doğru Kullanmak
Eleştirel Düşünmenin Önemi
“AI ile Öğrenmek” vs “AI’ye Bağımlı Olmak”
Günlük Hayatta Bilinçli Kullanım
Bilinçli AI Kullanımı İçin Pratik Öneriler
Bölüm Özeti
Soru Cevap

ETİK, RİSKLER VE GÜVENLİK

Algoritmik Önyargı
Veri Gizliliği
Deepfake ve Dezenformasyon
Yapay Zekâ Güvenliği
“Makine Karar Vermeli mi?” Sorusu
Etik ve Güvenlik İçin Örnek Senaryolar
Bölüm Özeti
Soru Cevap

YAPAY SİNİR AĞLARI VE DERİN ÖĞRENME

Yapay Sinir Hücresi (Perceptron)
Katmanlar ve Ağ Yapısı
Giriş Katmanı
Gizli Katmanlar
Çıkış Katmanı
Basit Perceptron ile Mantık Kapıları
Örnek: AND Kapısı
Derin Öğrenme Nedir?
Derin Öğrenmenin Öğrenme Süreci
Derin Öğrenmenin Uygulama Örnekleri
Derin Öğrenme ile El Yazısı Rakam Tanıma
Bölüm Özeti
Soru Cevap

ÜRETKEN YAPAY ZEKÂ VE BÜYÜK DİL MODELLERİ

Üretken Yapay Zekâ (Generative AI)
Büyük Dil Modelleri: Dilin Devleri
ChatGPT’nin Basitleştirilmiş Çalışma Mantığı
Metin, Görsel ve Müzik Üretimi
Halüsinasyon Problemi
Prompt Yazma Sanatı
Bölüm Özeti
Soru Cevap

TEACHABLE MACHINE (GOOGLE) – MAKİNEYİ ÖĞRETMEK ARTIK KOLAY

Duruş ve Hareket Kontrolü (Görüntü Tanıma)
Atık Ayrıştırma Rehberi (Nesne Tanıma)
Sesli Komutla Kontrol (Ses Tanıma)
Teachable Machine Nasıl Çalışır? (Basit Akış)
Adım Adım: Atık Ayrıştırma Yapay Zekası

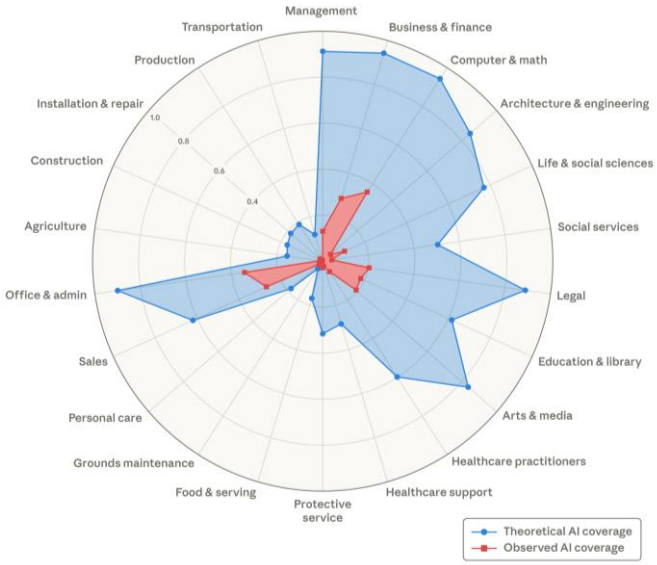
Teachable Machine ile Basit Bir Deney
Teachable Machine'in Özellikleri
Basit Kod Örneği ile Kullanım
Mini Uygulama: Teachable Machine ile Kendi Modelini Oluştur
Öğrenci Kılavuzu: Teachable Machine ile Kendi Modelini Kurma
Bölüm Özeti
Soru Cevap

TERİMLER SÖZLÜĞÜ

GİRİŞ

Yapay zekâ devrimi hakkında yapılan tartışmaların büyük bir kısmı, teknolojinin ne kadar hızlı geliştiği üzerine odaklanır. Ancak çoğu zaman gözden kaçan daha kritik bir soru vardır: *Bu teknolojiyi gerçekten ne kadar kullanıyoruz?* Anthropic tarafından 2026 yılında yayımlanan ve meslek gruplarına göre yapay zekânın teorik kapasitesi ile gözlemlenen kullanımı arasındaki farkı ortaya koyan grafik, bu soruya çarpıcı bir yanıt sunmaktadır.

Şekil 1. Yapay Zekânın Görünmeyen Gücü: Potansiyel ile Gerçek Kullanım Arasındaki Uçurum



Bu grafik, iki farklı dünyayı aynı anda gösterir. Birincisi, yapay zekânın teknik olarak neleri yapabileceğini temsil eden teorik kapasitedir. İkincisi ise insanların ve kurumların bu kapasiteyi ne ölçüde kullandığını gösteren gerçek kullanımdır. Grafik dikkatle incelendiğinde, bu iki alan arasında ciddi bir boşluk olduğu açıkça görülür. Başka bir deyişle, yapay zekâ çok şey yapabilmektedir; ancak biz henüz onun sunduğu imkânların küçük bir kısmını hayata geçirebilmiş durumdayız.

Bu durum özellikle beyaz yakalı mesleklerde daha belirgindir. Yönetim, finans, bilgisayar bilimleri, hukuk ve medya gibi alanlarda yapay zekânın teorik etkisi oldukça yüksektir. Bu mesleklerde yapılan işlerin önemli bir kısmı bilgi işleme, analiz, yazma ve karar verme gibi bilişsel süreçlere dayanır. Bu da onları yapay zekâ için son derece uygun hale getirir. Nitekim büyük dil modelleri ve diğer gelişmiş algoritmalar, bu tür görevleri başarıyla yerine getirebilecek kapasiteye ulaşmıştır. Ancak grafikte görüldüğü gibi, bu alanlarda gerçek kullanım hâlâ sınırlıdır. Kurumlar ve bireyler, teknolojinin sunduğu olanakları henüz tam anlamıyla içselleştirememiştir.

Buna karşılık tarım, inşaat, üretim ve bakım-onarım gibi fiziksel emeğe dayalı mesleklerde hem teorik kapasite hem de gerçek kullanım düşüktür. Bunun temel nedeni, bu işlerin yalnızca bilgi işleme ile değil, aynı zamanda fiziksel çevreyle etkileşim gerektirmesidir. Yapay zekâ, dijital dünyada son derece güçlü olsa da fiziksel dünyada hâlâ sınırlamalarla karşı karşıyadır. Robotik sistemlerin gelişimi bu açığı kapatmaya çalışsa da, bu alanlarda dönüşüm daha yavaş ilerlemektedir.

Grafiğin en çarpıcı yönü ise, teorik kapasite ile gerçek kullanım arasındaki boşluktur. Bu boşluk, aslında yapay zekâ devriminin henüz başlangıç aşamasında olduğunu göstermektedir. Bugün gördüğümüz etkiler, potansiyelin yalnızca küçük bir kısmını yansıtmaktadır. Eğer bu boşluk kapanırsa, yani gerçek kullanım teorik kapasiteye yaklaşırsa, çok daha büyük bir dönüşüm yaşanacaktır. Bu dönüşüm yalnızca verimlilik artışı ile sınırlı kalmayacak; aynı zamanda iş yapma biçimlerini, organizasyon yapılarını ve hatta meslek tanımlarını kökten değiştirecektir.

Bu noktada önemli bir soru ortaya çıkar: Neden bu boşluk var? Yapay zekâ bu kadar güçlü ise neden daha yaygın kullanılmıyor? Bunun birkaç temel nedeni vardır. İlk olarak, güven sorunu önemli bir engeldir. İnsanlar, nasıl çalıştığını tam olarak anlamadıkları sistemlere karşı temkinlidir. Bu durum, yapay zekânın "kara kutu" olarak algılanmasıyla doğrudan ilişkilidir. İkinci olarak, organizasyonel direnç söz konusudur. Kurumlar, mevcut iş süreçlerini değiştirmekte isteksiz davranabilir. Üçüncü olarak ise bilgi ve yetkinlik eksikliği önemli bir faktördür. Pek çok kişi, yapay zekâyı nasıl etkili bir şekilde kullanacağını henüz bilmemektedir.

Bu bağlamda, yapay zekâ alanında asıl rekabet avantajı, yalnızca daha iyi modeller geliştirmekten değil, bu modelleri gerçek dünyada etkin bir şekilde kullanabilmekten geçmektedir. Teknolojiyi üretmek ile onu uygulamak arasında ciddi bir fark vardır. Geleceğin kazananları, bu iki alanı birleştirebilenler olacaktır.

Bu grafik, aynı zamanda eğitim ve meslek seçimi açısından da önemli ipuçları sunar. Analitik düşünme, veriyle çalışma ve soyut problem çözme gerektiren meslekler, yapay zekâdan en fazla etkilenecek alanlar arasında yer almaktadır. Bu nedenle bireylerin kendilerini yalnızca teknik becerilerle değil, aynı zamanda yapay zekâ ile birlikte çalışabilme yetkinliğiyle donatmaları gerekmektedir.

Sonuç olarak, bu grafik bize şunu söylemektedir: Yapay zekâ devrimi, sandığımızdan çok daha büyük, ancak henüz tam olarak gerçekleşmemiş bir dönüşümdür. Bugün gördüğümüz değişim, aslında gelecekte yaşanacak olanın bir ön izlemesidir. Asıl soru, yapay zekânın ne yapabileceği değil; bizim onu ne kadar hızlı ve ne kadar akıllıca kullanabileceğimizdir.

Merhaba Yapay Zekâ

Sabah uyandınız. Eliniz neredeyse refleks gibi telefona uzandı. Alarmı kapattınız, ama ekranı kapatmadınız. Gün daha başlamadan, dünya çoktan akmaya başlamıştı.

Ekranı açtığınızda sosyal medya uygulaması size tam da ilginizi çekecek içerikleri sıraladı. Sanki dün aklınızdan geçenleri biliyormuş gibi… Birkaç dakika sonra müzik uygulaması, ruh halinize

şaşırtıcı biçimde uyan bir şarkı önerdi. Ne çok ne az; tam kararında. Evden çıkarken navigasyon uygulaması trafiği analiz etti ve size en hızlı yolu gösterdi. Üstelik sadece en kısa yolu değil, en akıllı yolu.

Bütün bunlar birkaç saniye içinde oldu.

Siz henüz tam olarak uyanmamışken, arka planda çalışan sistemler çoktan milyonlarca küçük hesaplama yapmıştı. Hangi videoyu izleyeceğinizi, hangi şarkıyı dinleyeceğinizi, hangi yolu kullanacağınızı belirleyen görünmez bir süreç işlemekteydi.

Peki hiç düşündünüz mü?

Bu sistemler sizi sizden daha iyi nasıl tanıyor?

Bu sorunun cevabı ilk bakışta biraz rahatsız edici olabilir. Çünkü cevap sadece teknolojiyle ilgili değil; aynı zamanda davranışlarımızla, alışkanlıklarımızla ve hatta tercih ettiğimizi sandığımız seçimlerle ilgilidir. Gün içinde attığınız her adım, tıkladığınız her bağlantı, izlediğiniz her video küçük bir iz bırakır. Bu izler tek başına önemsiz gibi görünür, ama bir araya geldiklerinde anlamlı bir tablo oluşturur. İşte bu tabloyu okuyan, yorumlayan ve tahminlerde bulunan sistemlere biz *yapay zekâ* diyoruz.

Yapay zekâ, çoğu zaman gözle görülmez. Bir robot gibi karşınıza çıkmaz, sizinle yüz yüze konuşmaz. Ama etkisi, gündelik hayatınızın her anında hissedilir. Bazen bir öneri olarak, bazen bir uyarı olarak, bazen de bir kolaylık olarak karşınıza çıkar. Sizi yönlendirmez gibi görünür, ama seçeneklerinizi şekillendirir.

Düşünün: Bir video platformunda karşınıza çıkan ilk içerik, gerçekten sizin seçiminiz midir? Yoksa sizin için seçilmiş bir olasılık mıdır? Bir müzik listesinde dinlediğiniz şarkılar, sizin zevkinizin sonucu mu, yoksa zevkinizin yönlendirilmiş bir yansıması mı? Bu soruların kesin cevapları yoktur. Ama şunu söyleyebiliriz: Artık sadece biz makineleri kullanmıyoruz; makineler de bizim davranışlarımızı anlamaya çalışıyor.

Bu noktada önemli bir şeyi fark etmek gerekir. Yapay zekâ sihirli bir varlık değildir. O, düşünen bir beyin ya da hisseden bir bilinç değildir. Daha çok, çok büyük miktarda veriyi işleyebilen ve bu verilerden anlamlı sonuçlar çıkarabilen bir sistemdir. Yani sizi “tanıdığı” için değil, sizin gibi milyonlarca insanın davranışlarını analiz ettiği için size doğru tahminlerde bulunur. Büyük dil modelleri (LLM’ler), yapay zekâ alanının bir alt dalı olan doğal dil işleme (NLP) kapsamında geliştirilmiş sistemlerdir. İnsan dilini anlayıp metin üretebildikleri için, yapay zekânın dil temelli yeteneklerini temsil eden en gelişmiş örnekler arasında yer alırlar.

> “Kara kutu”, bir sistemin girdi ve çıktılarının bilindiği, ancak içinde neler olup bittiğinin bilinmediği ya da anlaşılamadığı durumları tanımlayan bir kavramdır. Büyük Dil Modelleri (Large Language Models), özellikle derin öğrenme temelli oldukları için: milyarlarca parametre içerir, çok başarılı sonuçlar üretir ama “Bu cevabı neden verdi?” sorusu çoğu zaman açık değildir.

Başka bir deyişle, yapay zekâ sizi özel olarak bilmez; ama sizin gibileri çok iyi tanır. Bu kitapta birlikte yapacağımız yolculuk tam olarak bu noktadan başlıyor. Amacımız, yapay zekâyı bir “kara kutu” olmaktan çıkarmak. Onu gizemli bir teknoloji olmaktan ziyade, anlaşılabilir bir araç haline getirmek. Çünkü bir şeyi anlamaya başladığınızda, ondan korkmazsınız. Aksine, onu kullanmayı öğrenirsiniz.

Önümüzdeki bölümlerde şu soruların peşine düşeceğiz: Yapay zekâ gerçekten nasıl çalışır? Nasıl öğrenir? Nerelerde kullanılır? Ve en önemlisi, bizim hayatımızı nasıl değiştirir? Ama her şeyden önce, onu tanımamız gerekiyor. Çünkü içinde yaşadığımız dünya hızla değişiyor. Ve bu değişimin merkezinde, çoğu zaman fark etmeden etkileşim kurduğumuz bu “görünmez zekâ” yer alıyor. Artık yalnız değiliz. Kararlarımızın etrafında, sessizce çalışan başka bir akıl daha var. İşte bu bölüm, o akılla tanıştığınız yer.

🧠 Yapay Zekâ Nedir, Ne Değildir?

“Yapay zekâ” dendiğinde çoğu insanın zihninde iki güçlü imge belirir: İnsan gibi düşünen robotlar ve bilim kurgu filmlerindeki o gizemli makineler. Oysa gerçek, bu imgelerden hem daha sade hem de daha ilginçtir. Yapay zekâ, görünürde dramatik değildir; ama etkisi derindir. Sessiz çalışır, gözle görülmez, fakat kararlarımızın etrafında dolaşır.

En basit haliyle yapay zekâyı şöyle tanımlayabiliriz:

Yapay zekâ, makinelerin normalde insan zekâsı gerektiren görevleri yapabilmesidir.

Bu görevler düşündüğümüzden daha geniştir. Bir fotoğraftaki nesneleri tanımak, bir cümleyi başka bir dile çevirmek, bir e-postayı “önemli” ya da “önemsiz” olarak ayırmak, bir soruya anlamlı bir cevap vermek… Bunların hepsi, geçmişte yalnızca insanlara özgü olduğunu düşündüğümüz becerilerdi. Bugün ise makineler bu görevlerin çoğunu oldukça başarılı bir şekilde yerine getirebiliyor.

Fakat burada dikkat edilmesi gereken kritik bir nokta vardır: Yapay zekâyı gerçekten anlamak istiyorsak, sadece ne yaptığını değil, ne yapmadığını da açıkça görmemiz gerekir.

Yapay zekâ bilinçli değildir. Yani kendisinin farkında olan bir varlık değildir. “Ben varım” demez, diyemez. Yapay zekânın duyguları yoktur. Üzülmez, sevinmez, korkmaz. Bir metni okuduğunda ya da bir görüntüyü analiz ettiğinde, gerçek anlamda anladığı söylenemez.
Ve belki de en önemlisi, kendi başına amaç belirlemez. Ona verilen hedefler doğrultusunda çalışır.

Bu noktada bir örnek düşünelim. Bir yapay zekâya bir soru sordunuz ve size şu cevabı verdi: “Seni anlıyorum.” Bu cümle kulağa oldukça insani gelir. Hatta bazı durumlarda ikna edici bile olabilir. Ama gerçekte olan şey çok daha farklıdır. Yapay zekâ burada sizi “anladığı” için değil, daha önce gördüğü milyonlarca benzer örnek arasında bu cümlenin uygun bir cevap olduğunu “tahmin ettiği” için bu ifadeyi kullanır. Yani ortada bir duygu, bir bilinç ya da bir içsel deneyim yoktur.
Sadece son derece gelişmiş bir tahmin sistemi vardır.

Bu durumu bir bakıma şöyle düşünebiliriz: Yapay zekâ, büyük bir kütüphanede dolaşan çok hızlı bir okuyucu gibidir. Bu kütüphanede milyarlarca cümle, görüntü ve örnek bulunur. Siz bir soru sorduğunuzda, o da bu devasa kütüphanede çok hızlı bir tarama yapar ve en uygun cevabı bulur. Ama bu cevap, bir “anlama” sürecinin değil, bir eşleştirme ve olasılık hesaplama sürecinin sonucudur.

Burada önemli bir fark ortaya çıkar: İnsanlar dünyayı deneyimler, hisseder ve anlamlandırır. Yapay zekâ ise dünyayı *veri olarak görür.* İnsan için bir fotoğraf bir anı olabilir; yapay zekâ için ise o fotoğraf, belirli renklerin ve şekillerin matematiksel bir düzenidir.

Bu farkı anlamak, yapay zekâya bakış açımızı kökten değiştirir. Çünkü böylece onu olduğundan daha “insan gibi” görme hatasından kaçınırız. Yapay zekâyı abartmak da, küçümsemek de yanıltıcıdır. Doğru yaklaşım, onu olduğu gibi görmekten geçer: Ne sihirli bir varlık ne de basit bir araç; ikisinin arasında, güçlü ama sınırlı bir sistem.

Bir başka önemli nokta da şudur: Yapay zekâ kendi kendine ortaya çıkmaz. Onu insanlar tasarlar, eğitir ve yönlendirir. Hangi verilerle besleneceğine, neyi öğrenip neyi öğrenmeyeceğine insanlar karar verir. Bu yüzden yapay zekânın yaptığı hatalar çoğu zaman onun değil, onu geliştiren sistemlerin bir yansımasıdır.

Sonuç olarak şunu söyleyebiliriz:

Yapay zekâ düşündüğümüz gibi “anlayan” bir zihin değildir; ama düşündüğümüzden çok daha iyi “tahmin eden” bir sistemdir.

Ve belki de onu bu kadar güçlü yapan şey tam olarak budur.

Zekâ Nedir? İnsan ve Makine

Zekâ dediğimiz kavram, ilk bakışta çok tanıdık görünür. Hepimiz “zeki” birini tarif edebiliriz: hızlı düşünen, doğru kararlar veren, zor problemleri çözebilen biri. Ama biraz durup düşündüğümüzde, zekânın aslında ne kadar karmaşık bir şey olduğunu fark ederiz. Zekâ sadece bilgi sahibi olmak değildir; aynı zamanda o bilgiyi *doğru yerde, doğru zamanda kullanabilme yeteneğidir.*

Bir insan zekâsını ele alalım. İnsan, yeni bir durumu anlayabilir, geçmiş deneyimlerinden öğrenebilir ve daha önce hiç karşılaşmadığı bir probleme çözüm üretebilir. Üstelik bunu tek bir alanda değil, birçok farklı alanda yapabilir. Sabah matematik problemi çözen bir öğrenci, öğleden sonra mutfakta yeni bir tarif deneyebilir, akşam ise bir hikâye yazabilir. Bu esneklik, insan zekâsının en dikkat çekici özelliklerinden biridir.

Yapay zekâ alanında son yıllarda en çok dikkat çeken kavramlardan biri, modellerin büyüklüğünü ifade eden “parametre sayısı”dır. Özellikle büyük dil modelleri söz konusu olduğunda, “milyar” ve hatta “trilyon” gibi sayılar artık sıradan hale gelmiştir. Peki bir dil modelinde *1 trilyon parametre* bulunması gerçekten ne anlama gelir? Bu sayı yalnızca teknik bir büyüklük müdür, yoksa modelin zekâsı hakkında bize daha derin bir şeyler mi söyler?

Bir dil modelindeki parametreler, en basit ifadeyle modelin *öğrenilmiş hafızasını* temsil eder. Bunlar, eğitim süreci boyunca veriden çıkarılan ve sürekli güncellenen sayısal değerlerdir. Her bir parametre, modelin belirli bir kelimeye, ifadeye ya da dilsel örüntüye ne kadar önem vereceğini belirleyen küçük bir ayar düğmesi gibidir. Tek başına bir parametrenin anlamı yoktur; ancak milyarlarcası bir araya geldiğinde, dilin karmaşık yapısını temsil edebilen bir sistem ortaya çıkar.

Bu açıdan bakıldığında, 1 trilyon parametreye sahip bir model, son derece geniş ve derin bir temsil kapasitesine sahiptir. Böyle bir model, yalnızca kelimeleri değil; kelimeler arasındaki ince ilişkileri, bağlamı, tonlamayı, hatta kimi zaman kültürel çağrışımları bile yakalayabilir. İnsan dilinin çok katmanlı doğası düşünüldüğünde, bu tür bir kapasiteye ulaşmak için büyük sayıda parametreye ihtiyaç duyulması şaşırtıcı değildir. Dil, basit bir semboller dizisi değil; anlamın, bağlamın ve niyetin iç içe geçtiği karmaşık bir yapıdır.

Bu durumu bir benzetmeyle açıklamak mümkündür. Az sayıda parametreye sahip bir model, sınırlı bir kelime hazinesine sahip bir öğrenciye benzer. Daha fazla parametre ise daha geniş bir kelime bilgisi, daha güçlü bir kavrayış ve daha esnek bir ifade yeteneği anlamına gelir. Bir trilyon parametreye ulaşıldığında ise model, adeta devasa bir kütüphaneye erişimi olan, farklı konular arasında bağlantı kurabilen bir zihin gibi davranmaya başlar. Ancak burada önemli bir nokta vardır: Büyük bir kütüphaneye sahip olmak, o bilgiyi her zaman doğru kullanmak anlamına gelmez.

Gerçekten de, model büyüklüğü ile performans arasında güçlü bir ilişki olsa da, bu ilişki mutlak değildir. Daha fazla parametre genellikle daha iyi sonuçlar üretir; ancak bu yalnızca belirli koşullar altında geçerlidir. Eğitim verisinin kalitesi, modelin mimarisi ve öğrenme sürecinin nasıl

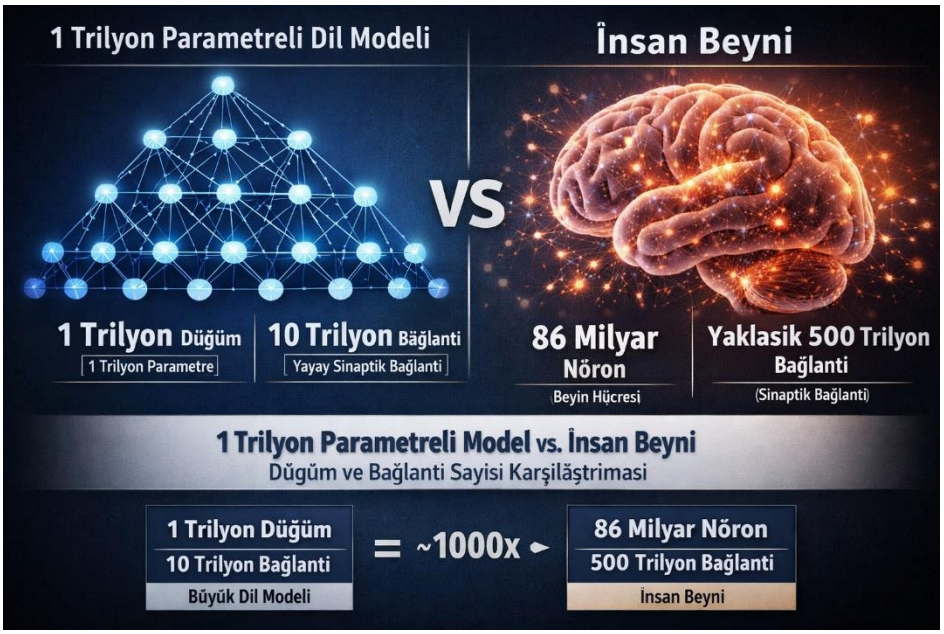
tasarlandığı en az parametre sayısı kadar önemlidir. Kötü veriyle eğitilmiş büyük bir model, küçük ama iyi eğitilmiş bir modelden daha kötü performans gösterebilir. Bu nedenle modern yapay zekâ araştırmaları, yalnızca modelleri büyütmeye değil, aynı zamanda onları daha verimli ve daha akıllı hale getirmeye odaklanmaktadır.

Bir trilyon parametrelik modellerin bir diğer önemli yönü ise maliyetleridir. Bu tür modelleri eğitmek ve çalıştırmak, son derece yüksek hesaplama gücü gerektirir. Bu da ciddi enerji tüketimi, güçlü donanım ihtiyacı ve yüksek ekonomik maliyet anlamına gelir. Dolayısıyla bu büyüklükteki modeller yalnızca teknik bir başarı değil, aynı zamanda altyapı ve kaynak meselesidir. Bu durum, yapay zekâ alanında büyük teknoloji şirketlerinin neden bu kadar belirleyici olduğunu da açıklar.

Parametre sayısının artmasıyla birlikte ortaya çıkan bir diğer mesele ise **anlaşılabilirliktir**. Model büyüdükçe, içinde neler olup bittiğini anlamak zorlaşır. Bu da yapay zekâ sistemlerinin "kara kutu" olarak nitelendirilmesine yol açar. Bir modelin doğru sonuç vermesi, onun nasıl çalıştığını anladığımız anlamına gelmez. Bu nedenle günümüzde, büyük modellerin yalnızca güçlü değil, aynı zamanda açıklanabilir olması gerektiği de giderek daha fazla vurgulanmaktadır.

Sonuç olarak, bir dil modelinde 1 trilyon parametre bulunması, yalnızca büyüklüğü değil, aynı zamanda potansiyeli ifade eder. Bu potansiyel, insan dilinin karmaşıklığını yakalama, farklı bağlamları anlama ve zengin metinler üretme yeteneğinde kendini gösterir. Ancak bu güç, beraberinde maliyet, karmaşıklık ve açıklanabilirlik gibi önemli soruları da getirir. Yapay zekânın geleceği, yalnızca daha büyük modeller inşa etmekte değil; bu modelleri daha verimli, daha anlaşılır ve daha erişilebilir hale getirmekte yatmaktadır.

Şekil 2. İnsan Beyni ile 1 Trilyon Parametreli Dil Modeli Karşılaştırması



İnsan Beyni ile Benzerlikler ve Farklar

Yapay sinir ağları, insan beyninden ilham alır, fakat tamamen aynı değildir.

Benzerlikler:

- Her iki sistem de nöronlardan oluşur.
- Öğrenme, ağırlıkların değiştirilmesi ile gerçekleşir. İnsan beyninde sinapsların gücü değişir, yapay ağlarda ağırlıklar güncellenir.
- Girdi → İşlem → Çıktı mantığı ortaktır.

Farklar:

- İnsan beyni enerji açısından çok verimlidir; yapay ağlar daha fazla elektrik gücü harcar.
- İnsan beyni karmaşık soyutlama ve sezgi yapabilir; yapay ağlar yalnızca verilerle sınırlıdır.
- İnsan beyninde öğrenme sürekli ve esnektir, yapay ağlarda ise genellikle eğitilmiş modele dayanır ve yeni verilere adapte olmak sınırlıdır.
- Beynimiz milyonlarca nöron ve milyarlarca bağlantıya sahiptir, yapay ağlar buna kıyasla küçüktür.

Günlük hayattan örnekler:

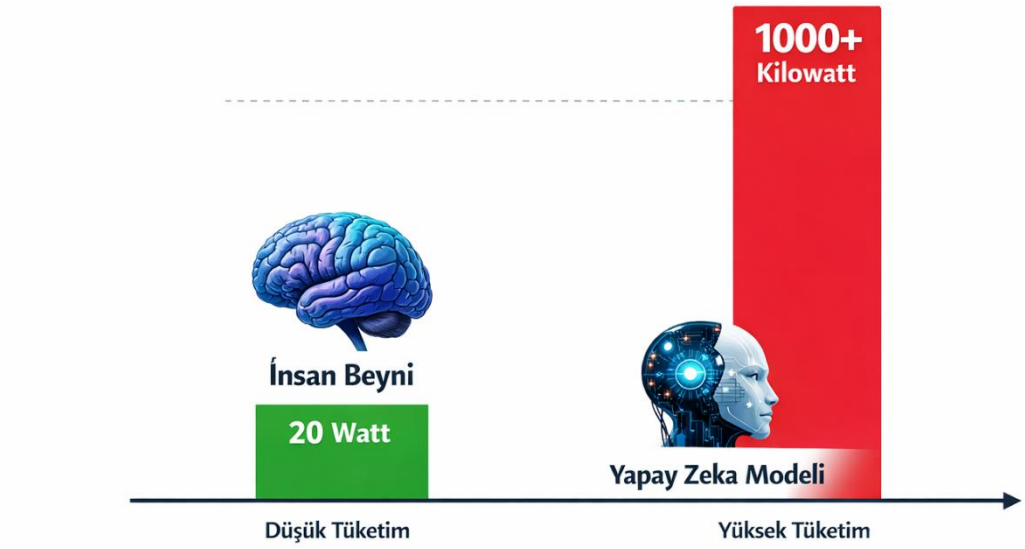
- İnsan beyni, bir yüzü hatırlamak için bağlamı, ses tonunu ve ifadeyi kullanır. Yapay ağlar ise genellikle sadece görüntüyü analiz eder.
- İnsan çocuk dil öğrenirken küçük örneklerden anlam çıkarır; yapay ağlar ise büyük veri setleri ile eğitilmelidir.

İnsan zekâsı aynı zamanda bağlamı kavrayabilir. Bir cümlenin anlamı, sadece kelimelerden değil, o cümlenin söylendiği durumdan, tonundan ve hatta geçmiş deneyimlerden etkilenir. İnsan, bu karmaşık ilişkileri çoğu zaman farkında bile olmadan çözebilir. Bir şakanın neden komik olduğunu, bir bakışın ne anlama geldiğini ya da bir hikâyenin neden etkileyici olduğunu anlayabilir.

Peki ya makineler?

Makineler de bir tür “zekâ” sergiler, ama bu zekâ insan zekâsından oldukça farklıdır. Makineler inanılmaz bir hızla hesap yapabilir. Bir insanın saatler sürecek bir işlemi, bir makine saniyeler içinde tamamlayabilir. Ayrıca devasa veri yığınları içinde gizli kalmış örüntüleri ortaya çıkarma konusunda son derece başarılıdırlar. Örneğin milyonlarca fotoğrafı analiz ederek bir kediyi tanımayı öğrenebilirler ya da milyonlarca metni inceleyerek bir cümlenin nasıl devam etmesi gerektiğini tahmin edebilirler.

Şekil 3. Beyin ile Yapay Modellerin Güç Tüketimi Karşılaştırması



Şekil 3, insan beyni ile yapay zekâ modelleri arasındaki *enerji verimliliği farkını* çarpıcı biçimde ortaya koymaktadır. İnsan beyni yaklaşık *20 watt* gibi son derece düşük bir güç tüketimiyle çalışırken, büyük ölçekli yapay zekâ modelleri *1000 watt ve üzeri* enerji gerektirebilmektedir. Bu fark, yalnızca niceliksel değil, aynı zamanda niteliksel bir ayrımı da yansıtır.

İnsan beyni, milyonlarca yıllık evrim sürecinde *enerji açısından son derece optimize edilmiş* bir sistemdir. Nöronlar arasındaki iletişim, minimum enerjiyle maksimum bilgi işleme kapasitesi sağlayacak şekilde düzenlenmiştir. Buna karşılık yapay zekâ modelleri, özellikle derin öğrenme tabanlı sistemler, çok büyük veri merkezlerinde çalışan ve yüksek hesaplama gücü gerektiren yapılardır. Bu nedenle ciddi miktarda elektrik tüketirler.

Grafik aynı zamanda yapay zekânın mevcut sınırlamalarına da işaret eder. İnsan beyni yalnızca az enerji tüketmekle kalmaz; aynı zamanda öğrenme, genelleme ve adaptasyon konularında da oldukça etkilidir. Yapay modeller ise benzer performanslara ulaşabilmek için çok daha fazla enerjiye ihtiyaç duyar.

Sonuç olarak bu karşılaştırma, yapay zekânın gücüne rağmen *enerji verimliliği açısından hâlâ insan beyninin gerisinde olduğunu* ve gelecekteki araştırmaların daha verimli sistemler geliştirmeye odaklanması gerektiğini göstermektedir.

Bunun yanında makineler, tekrarlı görevlerde neredeyse hiç hata yapmaz. Aynı işlemi binlerce kez aynı doğrulukla gerçekleştirebilirler. Bu da onları özellikle üretim, analiz ve otomasyon gibi alanlarda vazgeçilmez hale getirir.

Ancak burada çok önemli bir fark ortaya çıkar:

İnsan zekâsı genel, yapay zekâ ise çoğunlukla özeldir.

Bu ne demek?

İnsan zekâsı, farklı alanlara uyum sağlayabilen bir yapıya sahiptir. Bir insan yeni bir beceriyi öğrenebilir, öğrendiklerini başka alanlara aktarabilir ve farklı problemler arasında bağlantılar kurabilir. Örneğin satranç oynamayı bilen biri, stratejik düşünme becerisini iş hayatında da kullanabilir.

Yapay zekâ ise genellikle belirli bir görev için tasarlanır. Bu yüzden "dar yapay zekâ" olarak adlandırılır. Bir sistem satranç oynamada olağanüstü başarılı olabilir. Hatta dünya şampiyonlarını bile yenebilir. Ama aynı sistem, bir şiirin duygusal derinliğini anlamakta ya da basit bir günlük konuşmayı yorumlamakta yetersiz kalabilir.

Bu durum, yapay zekânın zayıf olduğu anlamına gelmez. Aksine, belirli bir alanda son derece güçlü olduğu anlamına gelir. Ama bu güç, genellenebilir değildir. Yani bir alandaki başarı, başka bir alana otomatik olarak aktarılmaz.

İnsan ile makine arasındaki bu farkı bir benzetmeyle düşünebiliriz. İnsan, çok yönlü bir sporcu gibidir. Birçok farklı dalda ortalamanın üzerinde performans gösterebilir. Yapay zekâ ise tek bir dalda olimpiyat şampiyonu olan bir sporcuya benzer. Kendi alanında kusursuza yakındır, ama başka bir alanda aynı başarıyı gösteremez.

Bir başka önemli fark da öğrenme biçimlerinde ortaya çıkar. İnsanlar genellikle az sayıda örnekle öğrenebilir. Bir çocuk birkaç kez kedi gördükten sonra, yeni bir kediyi kolayca tanıyabilir. Yapay zekâ ise genellikle çok daha fazla veriye ihtiyaç duyar. Binlerce, hatta milyonlarca örnek üzerinden öğrenir.

Bütün bu farklara rağmen, insan ve makine zekâsı birbirini tamamlayabilir. İnsan yaratıcılığı, sezgisi ve bağlam anlayışı ile makinelerin hızını, doğruluğunu ve veri işleme kapasitesini bir araya getirmek, çok güçlü bir işbirliği yaratır.

Sonuç olarak şunu söyleyebiliriz:

Zekâ tek bir şey değildir; farklı biçimleri vardır.
İnsan zekâsı esnek ve geneldir.
Makine zekâsı hızlı, güçlü ama çoğunlukla özeldir.

Ve belki de geleceğin en önemli sorusu şudur:
Bu iki farklı zekâ türü birlikte nasıl çalışacak?

📱 Günlük Hayatta Yapay Zekâ

Yapay zekâ artık geleceğin belirsiz bir vaadi değil; tam anlamıyla bugünün görünmez altyapısıdır. Sabah uyandığınız andan gece uyuyana kadar, fark etseniz de etmeseniz de onunla sürekli temas halindesiniz. Üstelik bu etkileşim çoğu zaman o kadar doğal ve akıcıdır ki, arka planda çalışan sistemleri düşünme gereği bile duymayız.

En yaygın örneklerden biri öneri sistemleridir. Bir video platformuna girdiğinizde karşınıza çıkan içerikler ya da bir müzik uygulamasında size sunulan çalma listeleri rastgele değildir. Bu sistemler, geçmişte ne izlediğinizi, neyi beğendiğinizi, ne kadar süreyle dinlediğinizi analiz eder. Ardından oldukça basit ama etkili bir varsayım yapar: *"Bu kişi bunu sevdiyse, buna benzeyen şeyleri de sevebilir."*

Bu basit görünen fikir, aslında son derece karmaşık matematiksel modeller ve büyük veri analiziyle desteklenir. Sizin gibi davranan milyonlarca kullanıcıdan öğrenilen örüntüler, size özel bir deneyim oluşturur. Böylece her kullanıcı, aynı platformu kullansa bile farklı bir "dijital dünya" görür.

Navigasyon uygulamaları da yapay zekânın günlük hayattaki en somut örneklerinden biridir. Eskiden haritalar statikti; bir noktadan diğerine nasıl gidileceğini gösterirdi. Bugün ise navigasyon sistemleri sürekli değişen bir problemi çözmektedir. Trafik yoğunluğu, kazalar, yol çalışmaları, hatta diğer kullanıcıların anlık hareketleri analiz edilerek size en hızlı rota sunulur.

Bu şu anlama gelir: Siz yola çıktığınızda, aslında yalnız değilsiniz. Aynı anda binlerce sürücünün verisi işlenir ve sistem her saniye yeni bir karar verir. Bir anlamda, şehir bir "canlı organizma" gibi izlenir ve yönlendirilir.

Sosyal medya platformları ise yapay zekânın belki de en yoğun kullanıldığı alanlardan biridir. Karşınıza çıkan gönderiler, videolar ve reklamlar kesinlikle rastgele değildir. Her biri, karmaşık bir optimizasyon probleminin sonucudur. Bu problemin temel sorusu şudur: *"Bu kullanıcıyı ekranda biraz daha tutacak en iyi içerik hangisi?"*

Bu hedef doğrultusunda sistem, sizin davranışlarınızı sürekli öğrenir. Hangi gönderide durdunuz, hangisini hızlıca geçtiniz, neyi beğendiniz, neye yorum yaptınız… Tüm bu mikro davranışlar analiz edilerek bir profil oluşturulur. Ve bu profil, sürekli güncellenir. Yani siz değiştikçe, sistem de sizinle birlikte değişir.

Bununla birlikte yapay zekâ sadece eğlence veya sosyal medya ile sınırlı değildir. E-posta kutunuzdaki spam filtreleri, gereksiz mesajları ayıklamak için yapay zekâ kullanır. Online alışveriş siteleri, ilginizi çekebilecek ürünleri tahmin eder. Bankacılık sistemleri, dolandırıcılık ihtimallerini tespit etmek için işlemleri analiz eder. Hatta akıllı telefonunuzun yüz tanıma özelliği bile bir yapay zekâ uygulamasıdır.

Tüm bu örneklerin ortak bir noktası vardır: *Görünmezlik.*

Yapay zekâ genellikle kendini göstermez. Arka planda çalışır, kararlar alır ve size sadece sonucu sunar. Bu yüzden çoğu zaman onun varlığını fark etmeyiz. Ama aslında o, dijital dünyayı şekillendiren temel güçlerden biridir.

Bu durum önemli bir soruyu da beraberinde getirir: Eğer karşımıza çıkan içerikler, gördüğümüz yollar ve hatta aldığımız öneriler yapay zekâ tarafından belirleniyorsa, ne kadar özgürüz?

Bu soru, yapay zekânın sadece teknik değil, aynı zamanda toplumsal ve etik bir konu olduğunu gösterir. Çünkü bu sistemler sadece hayatımızı kolaylaştırmaz; aynı zamanda seçimlerimizi de etkiler.

Sonuç olarak, yapay zekâ artık hayatımızın ayrılmaz bir parçasıdır. Onu anlamak, sadece teknolojiyi anlamak değil; aynı zamanda kendi davranışlarımızı, tercihlerimizi ve hatta kararlarımızı anlamak demektir.

Ve belki de en çarpıcı gerçek şudur: *Yapay zekâyı kullanırken, aslında o da bizi "öğrenmektedir."*

🧩 Yapay Zekâ Efsaneleri ve Gerçekler

Yapay zekâ, son yıllarda gündelik hayatımızın ayrılmaz bir parçası hâline gelmiş olsa da, teknoloji hakkındaki yanlış inanışlar hâlâ çok yaygındır. Bu yanlışlar, yapay zekâyı ya olduğundan çok daha güçlü ve akıllı gösterir, ya da gereksiz yere korkutucu bir tehdit gibi sunar. Oysa gerçek, çoğu zaman hayal gücümüzden çok daha ilginç ve öğreticidir. Bu bölümde, yapay zekâ hakkında en yaygın efsaneleri ve arkasındaki gerçekleri ele alacağız.

Efsane 1: "Yapay zekâ insan gibi düşünüyor"

Belki de en yaygın yanlış inanış, yapay zekânın insan zihni gibi düşündüğü yönündedir. Film ve dizilerde sıkça gördüğümüz, kendi kararlarını alabilen, hisleri olan ve bilinç sahibi robot imgesi, gerçek dünyadaki yapay zekâ ile karıştırılır.

Gerçek ise oldukça farklıdır: Yapay zekâ *düşünmez.* İnsan gibi duygulara sahip değildir, kendi iradesi yoktur ve bilinçli bir farkındalıkla karar vermez. Yapay zekânın yaptığı şey, yalnızca *verilerdeki örüntüleri tanımak ve bunlardan tahminler yapmaktır.* Örneğin bir dil modeli, sizin sorunuza verdiği cevabı, daha önce gördüğü milyonlarca örnekten en uygun olanı seçerek üretir. Ortada bir "anlama" yoktur; sadece karmaşık bir tahmin mekanizması vardır.

Efsane 2: "Yapay zekâ yakında dünyayı ele geçirecek"

Bir diğer yaygın korku, yapay zekânın kısa sürede insanlığı kontrol altına alacağıdır. Medyada sıkça duyduğumuz "robotlar işimizi alacak" veya "yapay zekâ insanlığı tehdit ediyor" gibi başlıklar, bu kaygıyı körükler.

Gerçek ise oldukça sınırlıdır: Günümüzde kullanılan yapay zekâ sistemleri *dar yapay zekâ* kategorisindedir. Yani her biri yalnızca belirli bir görevde uzmandır. Satrançta veya Go oyununda insan şampiyonları yenebilir, fakat aynı sistem günlük hayat problemlerini veya farklı bir alandaki yaratıcı görevleri çözemez. Bugün kullandığımız sistemler, dünyayı ele geçirmekten çok uzaktır; onlar, belirli problemleri çözmek için programlanmış araçlardır.

Efsane 3: "Yapay zekâ her şeyi bilir"

Bir diğer yanlış inanış, yapay zekânın "her şeyi bildiği" düşüncesidir. Özellikle internet ve bilgiye hızlı erişim yeteneği, insanların böyle bir izlenim kazanmasına yol açar.

Oysa gerçek şu ki, yapay zekâ *sadece eğitildiği verileri bilir.* Eğer bir sistem belirli bir dil, konu veya veri kümesi ile eğitildiyse, yalnızca o alanla sınırlı bilgi üretir. Bilinmeyen veya eğitim sırasında karşılaşılmamış durumlarda yapay zekâ *hatta yanıltıcı veya yanlış cevaplar verebilir.* Bu nedenle bir yapay zekâyı sınırsız bilgi kaynağı olarak görmek yanıltıcıdır.

Efsane 4: "Yapay zekâ hata yapmaz"

Belki de en tehlikeli yanlış, yapay zekânın hata yapamayacağına dair inançtır. Birçok insan, bilgisayarın veya yapay zekânın kesin doğrulukta çalıştığını varsayar.

Gerçek ise daha karmaşıktır: Yapay zekâ sistemleri, *veriye bağımlıdır.* Veri eksik, hatalı veya önyargılıysa, sonuçlar da yanlış olabilir. Örneğin bir görüntü tanıma sistemi, eğitildiği veride yeterli çeşitlilik yoksa bazı nesneleri yanlış tanıyabilir. Bir dil modeli, eğitim verisinde yanlış bilgiler varsa, bunları tekrar edebilir. Bu nedenle yapay zekâda hata yapma ihtimali her zaman vardır ve insanlar tarafından denetim ve doğrulama gereklidir.

Yapay zekâ hakkındaki efsaneler, çoğu zaman teknolojiyi yanlış anlamamızdan kaynaklanır. İnsan gibi düşünmediğini, dünyayı ele geçiremediğini, her şeyi bilmediğini ve hata yapabileceğini bilmek, hem onu daha etkili kullanmamızı sağlar hem de gereksiz korkuları ortadan kaldırır. Yapay zekâ, doğru anlaşıldığında *hayatımızı kolaylaştıran, bilgiye dayalı güçlü bir araçtır,* ama sihirli veya her şeyi bilen bir varlık değildir.

Düşün: Günlük yaşamınızda yapay zekâ ile etkileşiminizi gözden geçirin. Bu sistemlerin hangi önerileri sizin için doğru tahmin ediyor ve hangi durumlarda yanılabiliyor olabilir? Gerçek ile efsane arasındaki farkı fark etmek, bilinçli bir teknoloji kullanıcısı olmanın ilk adımıdır.

⚖️ Dar Yapay Zekâ ve Genel Yapay Zekâ

Yapay zekâyı anlamanın en etkili yollarından biri, sistemleri *iki temel kategoriye ayırmaktır:* dar yapay zekâ ve genel yapay zekâ. Bu sınıflandırma, teknolojiyi daha net kavramamızı sağlar ve hem günlük hayatımızda karşılaştığımız uygulamaları hem de bilim dünyasında tartışılan teorik kavramları ayırt etmemize yardımcı olur.

Dar Yapay Zekâ (Narrow AI)

Günümüzde kullandığımız tüm yapay zekâ sistemleri *dar yapay zekâ* kategorisine girer. Bu tür yapay zekâlar, belirli bir görev için tasarlanmıştır ve yalnızca o görevde yüksek performans gösterir. Örneğin, bir satranç oyunu için geliştirilmiş bir yapay zekâ, hamleleri hesaplamakta ve rakibi yenmekte olağanüstü başarılıdır. Ancak aynı sistem bir şiirin anlamını çözemez veya bir mutfakta yemek yapamaz.

Dar yapay zekâ örnekleri hayatımızın hemen her alanına girmiştir:

- **Yüz tanıma sistemleri**: Akıllı telefonlarda veya güvenlik kameralarında kullanılır. Fotoğrafları analiz eder, yüzleri tanır ve doğruluk oranı oldukça yüksektir. Ancak bu sistem, başka bir görevde çalıştırıldığında başarısız olur; örneğin bir resmi açıklayıcı bir şekilde yorumlayamaz.

- **Metin üreten modeller**: Haber özetleri, sosyal medya içerikleri veya e-posta yanıtları oluşturabilir. Yine de bu sistemler, bir şiirin estetik değerini veya duygusal anlamını gerçekten anlayamaz.
- **Öneri sistemleri**: Müzik, video veya alışveriş uygulamalarında kullanılır. Kullanıcının önceki tercihlerini analiz ederek önerilerde bulunur. Ancak başka bir bağlamda, örneğin kullanıcıya kişisel yaşam koçluğu yapmak gibi bir görevi yerine getiremez.

Dar yapay zekânın temel özelliği *uzmanlıkt*ır. Bu sistemler bir konuda olağanüstü başarılıdır, fakat uzmanlık alanları dışındaki işler için tamamen yetersizdir. Günlük hayatımızda karşılaştığımız çoğu yapay zekâ uygulaması bu kategoriye girer.

Genel Yapay Zekâ (Artificial General Intelligence – AGI)

Dar yapay zekânın aksine, bilim insanlarının uzun yıllardır hayalini kurduğu bir kavram daha vardır*: genel yapay zekâ.* Genel yapay zekâ, insan zekâsına daha yakın bir şekilde çalışması öngörülen sistemlerdir. Bu tür bir sistem:

- Farklı alanlarda düşünebilir,
- Yeni ve beklenmedik durumlara hızla uyum sağlayabilir,
- Öğrenmeyi öğrenebilir, yani kendi bilgi ve yeteneklerini geliştirebilir.

Başka bir deyişle, genel yapay zekâ, insan gibi *çok yönlü ve esnek* bir zekâya sahip olmalıdır. Bir insan, bir gün matematik problemi çözebilir, ertesi gün yemek yapabilir, başka bir gün yaratıcı bir hikâye yazabilir. Genel yapay zekânın amacı, bu esnekliği makinelerde gerçekleştirmektir.

Ancak bugün için böyle bir sistem *henüz mevcut değildir.* Bilim insanları, bilgisayar mühendisleri ve felsefeciler hâlâ AGI'nin nasıl geliştirilebileceğini tartışmaktadır. Teknik, etik ve felsefi engeller oldukça karmaşıktır. Örneğin, bir AGI'nin bilinç kazanıp kazanamayacağı veya kararlarında hangi etik ilkeleri kullanacağı hâlâ cevaplanmamış sorulardır.

Dar ve Genel Yapay Zekâ Arasındaki Farklar

- **Kapsam**: Dar yapay zekâ tek bir alanda uzmandır; genel yapay zekâ çok alanlıdır.
- **Esneklik**: Dar yapay zekâ belirli bir görevle sınırlıdır; genel yapay zekâ yeni durumlara uyum sağlayabilir.
- **Öğrenme**: Dar yapay zekâ yalnızca önceden programlandığı veya eğitildiği veri üzerinden öğrenir; genel yapay zekâ kendi öğrenme stratejilerini geliştirebilir.
- **Mevcudiyet**: Dar yapay zekâ günlük hayatımızda kullanılıyor; genel yapay zekâ hâlâ teorik ve deneysel bir konsepttir.

Dar yapay zekâ, hayatımızı kolaylaştıran ve belirli problemleri çözmek için tasarlanmış araçlar sunarken, genel yapay zekâ bilim dünyasının ve teknolojinin en büyük hedeflerinden biridir.

İlerleyen yıllarda AGI'nin ortaya çıkıp çıkmayacağı, hem teknoloji hem de etik açısından yeni tartışmalar başlatacaktır.

Düşün: Günümüzde kullandığınız yapay zekâ uygulamalarını düşünün. Bunlar hangi görevlerde başarılı, hangi görevlerde yetersiz? Eğer bir sistem hem sizin sosyal hayatınızı hem matematik problemlerinizi hem de yaratıcı projelerinizi yönetebilseydi, bu sizin hayatınızı nasıl değiştirirdi?

*Düşün:*Bir yapay zekâ gerçekten "anlayabilir" mi, yoksa sadece taklit mi eder? Sosyal medya sana mı hizmet ediyor, yoksa seni mi yönlendiriyor? İnsan zekâsı ile makine zekâsı birleşirse ortaya nasıl bir dünya çıkar?

🧪 *Deney*

Gün içinde kullandığın bir uygulamayı seç. Bu bir müzik uygulaması, video platformu ya da alışveriş sitesi olabilir.

Kendine şu soruları sor:
Bu uygulama bana ne öneriyor?
Bu önerileri neye göre yapıyor?
Benim hakkımda ne biliyor olabilir?

🤖 **Yapay Zekâ ile Dene**

Bir yapay zekâ aracına, örneğin ChatGPT, şu soruyu sor:

"Beni tanımadan bana bir film öner."

Ardından şu bilgiyi ekle:
"Ben bilim kurgu ve tarih seviyorum."

Aynı soruyu tekrar sor.

Cevapların nasıl değiştiğine dikkat et.
Aslında sen de küçük bir "öğrenme süreci" başlatmış oldun.

Bölüm Özeti

Bu bölümde, yapay zekâ kavramına ve günlük yaşamımızdaki kullanımına dair temel bilgileri inceledik. Yapay zekâ, basit bir tanımla *insan zekâsı gerektiren görevleri yerine getirebilen sistemler*dir. Bu görevler arasında görme, anlama, konuşma ve karar verme gibi yetenekler bulunur. Örneğin bir fotoğraftaki nesneleri tanımak, bir metni özetlemek veya bir soruya anlamlı cevap vermek yapay zekânın yapabileceği işler arasındadır.

Ancak burada çok kritik bir nokta vardır: Yapay zekâ *bilinçli değildir.* Bir yapay zekâ sistemi, sizle konuştuğunda ya da bir öneride bulunduğunda, aslında sizinle empati kurmaz veya sizi anlamaz; yalnızca geçmiş verilerden ve örüntülerden en uygun cevabı tahmin eder. Ortada

gerçek bir “anlama” yoktur; güçlü bir tahmin mekanizması vardır. Bu, yapay zekâyı doğru anlamak için en temel bilgidir.

Günlük hayatımızda yapay zekâyla fark etmeden etkileşim halindeyiz. Öneri sistemleri, sosyal medya akışları, müzik veya video önerileri, navigasyon uygulamaları ve akıllı asistanlar gibi birçok araç, her an bizim için veri toplar ve analiz eder. Bu sistemler, kullanıcı davranışlarını gözlemler, örüntüleri tanır ve gelecekteki tercihleri tahmin eder. Örneğin bir video platformu, izlediğiniz içeriklere bakarak yeni öneriler sunar; navigasyon uygulaması ise trafik durumunu, kazaları ve diğer sürücülerin hareketlerini analiz ederek en hızlı rotayı belirler.

Bütün bu sistemlerin *hepsi dar yapay zekâ* sınıfına girer. Dar yapay zekâ, yalnızca belirli bir görevde uzmanlaşmış sistemleri ifade eder. Bir satranç programı satrançta mükemmeldir, fakat bir şiirin anlamını çözemeyebilir. Bir yüz tanıma sistemi güvenlikte mükemmel çalışırken, başka bir görevi yerine getiremez. Günümüzde kullandığımız yapay zekâların tümü bu kategoriye aittir ve insan zekâsının esnekliğine sahip *genel yapay zekâ* hâlâ teorik bir hedef olarak bilim dünyasında tartışılmaktadır.

Bu bölümde ayrıca, yapay zekâ hakkında yaygın *yanlış inanışlar* da ele alındı. Örneğin, yapay zekânın insan gibi düşündüğü, her şeyi bildiği, dünyayı ele geçirebileceği veya hata yapamayacağı gibi inanışlar doğru değildir. Bu yanlış inanışlar, teknolojiyi ya olduğundan güçlü gösterir ya da gereksiz bir korku yaratır. Gerçek şudur: yapay zekâ, eğitildiği veriler kadar bilir ve hata yapabilir; bilinçli bir farkındalığı yoktur ve kendi başına amaç belirleyemez.

Bölümün bir diğer önemli mesajı, insan zekâsı ile yapay zekâ arasındaki farktır. İnsan zekâsı *genel ve esnektir*; yeni durumlara uyum sağlayabilir, farklı alanlarda öğrenebilir ve yaratıcı çözümler üretebilir. Yapay zekâ ise çoğunlukla **özel ve sınırlıdır**; yalnızca belirli görevlerde olağanüstü başarılı olur, ama insanın esnekliği ve yaratıcılığına sahip değildir.

Özetle, bu bölümün temel çıkarımları şunlardır:

1. *Yapay zekâ, insan zekâsı gerektiren görevleri yerine getirebilir,* ama bilinçli değildir.
2. *Günlük yaşamda birçok alanda kullanılır,* ancak çoğu zaman fark etmeden etkileşim halindeyiz.
3. *Tüm mevcut sistemler dar yapay zekâdır,* yani tek bir konuda uzmanlaşmışlardır.
4. *Yapay zekâ hakkında yaygın yanlış inanışlar vardır* ve bunları doğru anlamak, teknolojiyi bilinçli ve etkili kullanmamızı sağlar.
5. İnsan zekâsı esnek ve yaratıcıdır; yapay zekâ ise belirli görevlerde yüksek performans gösteren bir araçtır.

Soru-Cevap

Soru: Yapay zekâ (YZ) sistemleri bizi bizden daha iyi nasıl tanıyabilir?
Cevap: YZ sistemleri, çok büyük miktarda veriyi analiz ederek, davranış kalıplarımızı, tercih ve

alışkanlıklarımızı tespit edebilir. Örneğin internet aramalarımızı ve sosyal medya paylaşımlarımızı kullanarak öneriler sunabilir.

1. **Soru:** Yapay zekâ nedir?
   Cevap: Yapay zekâ, bilgisayarların insan benzeri zekâ gerektiren görevleri yerine getirebilmesini sağlayan teknolojidir. Örneğin konuşma tanıma, oyun oynama veya görsel tanıma gibi.

2. **Soru:** Yapay zekâ ne değildir?
   Cevap: YZ, insan gibi bilinçli, duygulu veya tamamen bağımsız karar verebilen bir varlık değildir. Yalnızca programlandığı şekilde ve veri ile çalışır.

3. **Soru:** Zekâ nedir ve insan zekâsıyla makine zekâsı nasıl farklıdır?
   Cevap: Zekâ, problem çözme, öğrenme ve mantık yürütme yeteneğidir. İnsan zekâsı duyguları, yaratıcılığı ve sezgiyi içerirken, makine zekâsı büyük veri ve algoritmalar üzerine kuruludur.

4. **Soru:** Günlük hayatta yapay zekâyı nerelerde görebiliriz?
   Cevap: Akıllı telefon asistanları, öneri sistemleri (Netflix, YouTube), online alışveriş siteleri ve navigasyon uygulamaları gibi birçok alanda YZ kullanılır.

5. **Soru:** Yapay zekâ hakkında en yaygın efsanelerden biri nedir?
   Cevap: “YZ insanları işsiz bırakacak ve dünyayı ele geçirecek” düşüncesi yaygındır. Gerçekte YZ çoğu zaman işleri kolaylaştırır ve insanla birlikte çalışır.

6. **Soru:** Dar yapay zekâ ve genel yapay zekâ arasındaki fark nedir?
   Cevap: Dar yapay zekâ, belirli bir görevi yapabilir (ör. satranç oynama), genel yapay zekâ ise insan gibi farklı görevleri öğrenip uygulayabilir.

7. **Soru:** “Yapay zekâ ile dene” başlığı altında ne yapılır?
   Cevap: Bu bölümde YZ uygulamalarıyla küçük deneyler yapılarak, sistemlerin nasıl çalıştığı gözlemlenir ve öğrenme süreçleri anlaşılır.

8. **Soru:** Bu bölümdeki ana amaç nedir?
   Cevap: Okuyucuya yapay zekânın temel kavramlarını, günlük hayatımıza etkilerini ve yaygın yanlış anlamaları tanıtmaktır.

9. **Soru:** Bölümün özeti nasıl özetlenebilir?
   Cevap: Yapay zekâ, insan zekâsını taklit eden ama bağımsız olmayan sistemlerdir. Günlük hayatımızda yaygındır, yanlış anlaşılmaları vardır ve dar ve genel yapay zekâ kavramlarıyla farklılık gösterir.

MAKİNE NASIL ÖĞRENİR? (TEMEL KAVRAMLAR)

Makine öğrenmesi, yapay zekânın en temel yapı taşlarından biridir. İnsan zekâsı gibi makineler de öğrenebilir mi? Evet, öğrenebilirler, ama insanın öğrenme süreciyle kıyaslandığında yöntemleri ve mantıkları oldukça farklıdır. İnsanlar deneyimlerinden ve gözlemlerinden öğrenirken, makineler verileri analiz ederek örüntülerden öğrenir ve tahminler üretir. Bu bölümde, makine öğrenmesinin mantığını, veriyle ilişkisini ve farklı öğrenme yöntemlerini örneklerle keşfedeceğiz. Matematiksel formüller yerine mantıksal işleyişi anlamaya odaklanacağız, böylece konuyu lise ve üniversite giriş seviyesinde herkes anlayabilir.

Makine öğrenmesinin temel dayanağı veridir. Veri olmadan makine öğrenmesi düşünülemez. Bu nedenle ilk olarak veriye ve onun makine öğrenmesindeki rolüne bakalım.

Veri: Yeni Dünyanın Yakıtı

Veri, yapay zekânın ve makine öğrenmesinin temel yakıtıdır. İnsan zekâsının düşünme ve öğrenme süreçlerinde deneyim ve gözlem gibi girdilere ihtiyaç duyması gibi, makineler de öğrenmek için verilere ihtiyaç duyar. Verinin önemi, günümüz dijital çağında giderek daha fazla artmaktadır. Akıllı telefonlarımız, sosyal medya hesaplarımız, e-ticaret alışkanlıklarımız ve hatta sağlık verilerimiz, modern yapay zekâ sistemlerinin çalışması için kritik birer kaynak haline gelmiştir. Bu bölümde verinin ne olduğu, türleri, nasıl toplandığı, temizlendiği, etiketlendiği ve büyük veri kavramı gibi temel konuları ele alacağız. Ayrıca veri kalitesinin önemi, önyargı sorunları ve basit uygulamalar üzerinden örnekler sunacağız.

Veri Nedir? Türleri

Veri, ham bilgi ve gözlemlerden oluşan, anlamlı bir şekilde işlenmeye hazır hâle getirilebilecek kayıtlar bütünüdür. Veri tek başına bir anlam taşımayabilir, ancak doğru analiz edildiğinde değerli bilgiye dönüşür. Yapay zekâ ve makine öğrenmesi bağlamında veri, algoritmaların öğrenmesi ve karar vermesi için kullanılan temel yapı taşıdır.

Veri türleri üç ana başlıkta incelenebilir:

1. **Metin Verisi (Text Data):** İnsan dilinde yazılmış her türlü bilgi metin verisi olarak kabul edilir. E-posta mesajları, sosyal medya gönderileri, haber makaleleri, blog yazıları veya sohbet kayıtları metin verisine örnektir. Örneğin, bir e-posta uygulamasında gelen kutunuza düşen “Promosyon: %50 indirim fırsatı!” mesajı, metin verisi olarak işlenir.

Metin verisi, NLP (Natural Language Processing) algoritmaları ile analiz edilebilir; spam filtreleme, duygu analizi ve çeviri gibi uygulamalarda kullanılır.

2. **Görüntü Verisi (Image Data):** Fotoğraflar, videolar ve taranmış belgeler gibi görsel materyaller, bilgisayarlı görü (Computer Vision) sistemlerinin eğitilmesinde kullanılır. Örneğin bir sağlık uygulamasında röntgen görüntüleri, makine öğrenmesi modellerine öğretilerek hastalık tespiti yapılabilir. Otomotiv sektöründe ise otonom araçlar, kameralar aracılığıyla yol üzerindeki nesneleri tanımak ve takip etmek için görüntü verisi kullanır.
3. **Ses Verisi (Audio Data):** İnsan konuşmaları, müzik, çevresel sesler ve telefon görüşmeleri gibi ses kayıtları, sesli asistanlar, konuşma tanıma sistemleri ve müzik öneri algoritmaları için veri kaynağıdır. Örneğin, Siri veya Alexa gibi asistanlar, kullanıcıların söylediklerini anlayabilmek için milyonlarca ses örneği üzerinde eğitilmiş modelleri kullanır.

Veri türleri sadece bunlarla sınırlı değildir. Sensör verileri, IoT cihazlarından gelen ölçümler, GPS konum bilgileri ve finansal veriler de makine öğrenmesi için kritik öneme sahiptir. Örneğin bir akıllı saat, kullanıcının kalp atış hızını ve adım sayısını sürekli kaydederek sağlık analizi yapılmasını sağlar.

Veri Toplama, Temizleme ve Etiketleme

Veri toplama süreci, makine öğrenmesinin ilk ve en temel adımıdır. Hangi verilerin toplanacağı, hangi kaynaklardan alınacağı ve hangi formatta olacağı belirlenir. Örneğin bir e-ticaret sitesi, kullanıcıların hangi ürünlere tıkladığını, hangi ürünleri sepete eklediğini ve hangi ürünleri satın aldığını kaydeder. Bu veriler, gelecekte kullanıcılara öneriler sunmak için kullanılır.

Toplanan veriler genellikle hamdır ve doğrudan model eğitimi için uygun değildir. Bu nedenle *veri temizleme* adımı önem kazanır. Temizleme işlemi, eksik verileri doldurma, hatalı girişleri düzeltme ve verileri standart formata getirme gibi işlemleri kapsar. Örneğin bir kullanıcı profilinde yaş bilgisi eksikse veya yanlış girilmişse modelin yanlış öğrenmesine neden olabilir.

Bir sonraki adım ise *veri etiketleme*dir. Etiketleme, modelin hangi veriyi hangi kategoriyle eşleştireceğini belirtir. Örneğin bir resim tanıma modelinde fotoğraflardaki nesneler “kedi”, “köpek” veya “araba” gibi etiketlerle işaretlenir. Bu, denetimli öğrenme (supervised learning) için kritik bir adımdır.

Python kodu çalıştırmak için birkaç seçenek vardır:

1. **IDLE (Python'un kendi editörü)**
    - Python'u bilgisayara kurduktan sonra gelir.
    - Başlat menüsünden IDLE (Python 3.x) açın.
    - Açılan pencerede "File → New File" diyerek yeni bir dosya oluşturun.
    - Kodunuzu buraya yazın ve "File → Save As" ile bir isim verip .py uzantısı ile kaydedin (örneğin meyve_veri.py).
    - Çalıştırmak için "Run → Run Module (F5)" tuşlayın.
2. **Jupyter Notebook** (önerilir, görselliği iyi)
    - Anaconda veya pip ile kurulabilir.
    - Tarayıcıda açılır, hücrelere kod yazılır ve Shift + Enter ile çalıştırılır.
    - Kod ile açıklamaları aynı sayfada göstermek çok kolaydır.
3. **Çevrimiçi Python editörleri**
    - replit.com, Google Colab gibi siteler.
    - Herhangi bir program kurmadan çalıştırabilir, kodu ve sonucu aynı anda görebilirsiniz.

Adım adım çalıştırma (Windows örneği)

1. Python yüklü değilse python.org sitesinden indirin ve kurun.
2. IDLE açın: Başlat → IDLE (Python 3.x)
3. "File → New File" ile yeni bir dosya açın.
4. Kodu kopyalayıp yapıştırın.
5. "File → Save As" ile örneğin meyve_veri.py olarak kaydedin.
6. "Run → Run Module (F5)" ile çalıştırın.

7. Çıktının IDLE penceresinde göründüğünü kontrol edin.

Diyelim ki elimizde *basit meyve resimleri* var ve bunları bilgisayarımızdaki bir klasöre koyduk. Python ile basit bir veri örneği ve etiketleme süreci şu şekilde gösterilebilir:

C:\Users\Necmi\Documents\Meyveler\
  elma1.jpg
  elma2.jpg
  muz1.jpg
  muz2.jpg

Python'da bu resimleri ve etiketlerini şöyle tanımlarız:

```
# Örnek veri: Basit meyve resimleri
images = ["elma1.jpg", "elma2.jpg", "muz1.jpg", "muz2.jpg"]

# Her resme karşılık gelen etiketler
labels = ["elma", "elma", "muz", "muz"]

# Kodun çalıştığını test etmek için yazdıralım
for i in range(len(images)):
  print(f"{images[i]} → {labels[i]}")
```

Çıktı:

```
elma1.jpg → elma
elma2.jpg → elma
muz1.jpg → muz
muz2.jpg → muz
```

Bu kod, veriyi ve etiketlerini birleştirerek modelin eğitilmesine hazır hâle getirir. Öğrenciler, farklı resim ve etiket kombinasyonları ekleyerek süreci test edebilir.

Büyük Veri Kavramı

Büyük veri (Big Data), geleneksel veri işleme yöntemlerinin yetersiz kaldığı çok büyük ve karmaşık veri kümelerini ifade eder. Günümüzde sosyal medya platformları, e-ticaret siteleri, sensörler ve IoT cihazları sürekli veri üretir. Bu veriler, depolanması, işlenmesi ve analiz edilmesi açısından klasik yöntemlerle yönetilemez.

Örneğin bir video paylaşım platformu, her gün milyonlarca video yüklenmesini ve izlenmesini kaydeder. Bu verilerin analiz edilmesi, kullanıcı davranışlarını anlamak, öneri algoritmalarını geliştirmek ve reklam hedeflemesi yapmak için gereklidir. Büyük veri, sadece hacim açısından

değil, aynı zamanda hız (velocity), çeşitlilik (variety) ve doğruluk (veracity) açısından da önem taşır.

Veri Kalitesi ve Önyargı

Veri kalitesi, modelin doğruluğunu ve güvenilirliğini doğrudan etkiler. Düşük kaliteli veya eksik veri, modelin yanlış öğrenmesine ve hatalı tahminler yapmasına neden olur. Örneğin, yüz tanıma sisteminde sınırlı bir demografik gruptan alınan veri, sistemin diğer grupları doğru tanıyamamasına yol açabilir. Bu, önyargı (bias) probleminin klasik bir örneğidir.

Veri toplarken ve işlerken önyargıların farkında olmak, modelin adil ve güvenilir sonuçlar üretmesi için kritik öneme sahiptir. Örneğin bir bankanın kredi tahmin sistemi, yalnızca belirli yaş ve gelir gruplarına ait verilerle eğitilirse, diğer gruplara karşı haksız değerlendirme yapabilir.

Basit Veri Örnekleri ve Mini Uygulamalar

Veri kavramını somutlaştırmak için birkaç basit örnek verelim:

---

1. **E-posta sınıflandırması:** Spam ve normal mailleri ayırmak için veri olarak e-postalar ve etiketler kullanılır. Python ile basit bir Naive Bayes sınıflandırıcı bu amaç için kullanılabilir.

```
# 1. Gerekli kütüphaneleri çağırıyoruz
from sklearn.feature_extraction.text import CountVectorizer  # Metinleri sayısal verilere dönüştürür
from sklearn.naive_bayes import MultinomialNB  # Basit ve hızlı bir sınıflandırma modeli
```

- CountVectorizer: Metinleri makine öğrenmesi için sayılara çevirir. Örneğin "Buy now" → [1,1,0,...] gibi kelime sayılarıyla temsil edilir.
- MultinomialNB: Naive Bayes algoritmasının, kelime sayıları gibi çoklu sayısal verilerle çalışan versiyonu. Spam tespiti gibi uygulamalarda çok kullanılır.

```
# 2. Veri ve etiketlerimizi tanımlıyoruz
emails = ["Buy now", "Meeting at 10", "Limited offer", "Project update"]
labels = [1,0,1,0]  # 1: spam, 0: normal
```

- emails: Örnek e-postalarımız.
- labels: Her e-posta için “spam” veya “normal” bilgisini tutar.
    - 1 → Spam
    - 0 → Normal

```
# 3. Metinleri sayısal verilere dönüştür
vectorizer = CountVectorizer()          # Vectorizer nesnesi oluştur
X = vectorizer.fit_transform(emails)    # emails listesindeki her cümleyi sayılara çevir
```

- fit_transform():
    1. Tüm e-postalarda hangi kelimeler var öğrenir (fit)
    2. E-postaları kelime sayıları ile temsil eder (transform)
- X artık makinenin anlayacağı *sayısal veri matrisi*dir.

```
# 4. Modeli oluştur ve eğit
model = MultinomialNB()  # Naive Bayes sınıflandırıcısı
model.fit(X, labels)     # Modeli X ve etiketlerle eğitiyoruz
```

- fit(): Model, hangi kelime kombinasyonlarının spam ile normal e-postalara karşılık geldiğini öğrenir.

```
# 5. Test e-postasını hazırla
test_email = ["Exclusive offer just for you"]
X_test = vectorizer.transform(test_email)  # Test e-postasını sayılara çevir
```

- Burada *fit_transform değil*, transform kullanıyoruz çünkü test verisi için modelin kelime hazinesini değiştirmek istemiyoruz.
- Test e-postası aynı kelime sayıları formatına dönüştürülür.

```
# 6. Tahmin yap
prediction = model.predict(X_test)
print("Spam mi?", "Evet" if prediction[0]==1 else "Hayır")
```

- predict(): Model, test e-postasının spam olup olmadığını tahmin eder.
- prediction[0]==1 → Eğer tahmin 1 ise spam, 0 ise normal.
- print() ile sonucu ekrana yazdırır.

🔷 Özet

1. E-postaları ve spam etiketlerini belirledik.
2. Metinleri sayısal verilere çevirdik (CountVectorizer).
3. Naive Bayes modeli ile eğittik.
4. Yeni bir e-postayı test ettik ve spam olup olmadığını tahmin ettik.

Çıktı örneği:

```
Spam mi? Evet
```

2. **Basit görüntü verisi:** El yazısı rakamları tanıma için MNIST veri seti kullanılabilir. Her resim bir veri örneği, rakam ise etikettir.

# Gerekli kütüphaneyi çağırıyoruz
from tensorflow.keras.datasets import mnist

- tensorflow.keras.datasets: İçinde hazır veri setleri vardır.
- mnist: El yazısı rakamlardan oluşan ünlü bir veri setidir (0–9 arası rakamlar).

MNIST veri setini yükle
(X_train, y_train), (X_test, y_test) = mnist.load_data()

- X_train → Eğitim için kullanılan resimler
- y_train → Bu resimlerin doğru rakam etiketleri
- X_test → Modeli test etmek için kullanılan resimler
- y_test → Test resimlerinin etiketleri

# Eğitim verisinin boyutunu ve bazı etiketleri göster
print("Eğitim verisi şekli:", X_train.shape)
print("Eğitim etiketleri (ilk 10):", y_train[:10])

- X_train.shape → (60000, 28, 28) gibi bir çıktı verir:
    - 60.000 resim, her biri 28x28 piksel
- y_train[:10] → İlk 10 resmin hangi rakam olduğunu gösterir.

Çıktı örneği:

Eğitim verisi şekli: (60000, 28, 28)
Eğitim etiketleri (ilk 10): [5 0 4 1 9 2 1 3 1 4]

---

3. **Ses verisi örneği:** Küçük bir ses veri seti ile basit konuşma tanıma modeli geliştirilebilir. Örneğin Python'da librosa kütüphanesi ile ses dalgaları analiz edilebilir.

# Gerekli kütüphaneyi çağır
import librosa

- librosa: Python'da ses dosyalarını analiz etmek için kullanılan bir kütüphane.

```
# 5 saniyelik bir ses dosyasını yükle
y, sr = librosa.load('konusma.wav', duration=5)
```

- 'konusma.wav' → Bilgisayarda bulunduğu klasörde ses dosyası
- y → Ses dalgasının sayısal değerleri (amplitüdler)
- sr → Örnekleme oranı, yani saniyede kaç veri noktası alındığı (genellikle 22050 veya 44100 Hz)

```
# Ses dalgasının ilk 10 değerini ve örnekleme oranını göster
print("Ses dalgası (ilk 10 örnek):", y[:10])
print("Örnekleme oranı:", sr)
```

Çıktı örneği:

```
Ses dalgası (ilk 10 örnek): [0.0012, 0.0035, -0.0021, ...]
Örnekleme oranı: 22050
```

- Ses dalgası bir liste gibi sayılardan oluşur.
- Bu sayılar, hoparlörden çıkan sesin titreşimlerini temsil eder.

Bu örnekler, verinin farklı türlerde nasıl kullanıldığını ve makine öğrenmesinin temel girdisi olduğunu gösterir.

Veri, sadece ham bilgi değil, aynı zamanda yapay zekânın öğrenme sürecinde kritik bir role sahiptir. Doğru veri toplama, temizleme ve etiketleme adımları, yüksek kaliteli veri ve önyargısız modellerin ön koşuludur. Büyük veri çağında, veri hacmi ve çeşitliliği arttıkça, doğru analiz ve yönetim yöntemleri daha da önem kazanır.

Özetle, veri; yapay zekânın yakıtıdır. Metin, görüntü ve ses verisi gibi farklı türlerde olabilir. Toplanması, temizlenmesi ve etiketlenmesi gereklidir. Büyük veri kavramı, günümüzün dijital dünyasında veri hacminin ve çeşitliliğinin önemini vurgular. Veri kalitesi ve önyargı, model performansını ve güvenilirliğini doğrudan etkiler. Basit uygulamalar üzerinden veri kullanımı ve işleme süreçlerini görmek, makine öğrenmesinin temel mantığını anlamak için kritik öneme sahiptir.

Bu bölüm, veri kavramını geniş kapsamlı şekilde ele alarak, öğrencilerin ve okuyucuların yapay zekânın temel yapı taşını anlamasına yardımcı olur ve gerçek dünyadaki uygulamalarla bağlantı kurmalarını sağlar.

Öğrenme Yöntemleri

Makine öğrenmesi, insan öğrenmesine bazı açılardan benzer ama kendi mantığı ve kuralları vardır. Temel olarak üç tür öğrenme yöntemi vardır: *denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme.* Her biri farklı problemlere uygundur ve farklı örneklerle anlaşılabilir.

Denetimli Öğrenme

Denetimli öğrenmede, makineye *hem girdi (input)* hem de doğru *çıktı (output)* gösterilir. Model, girdiler ile çıktılar arasındaki ilişkiyi öğrenir ve yeni veriler geldiğinde tahminler yapabilir.

- **Sınıflandırma örneği**: E-posta filtreleme sistemi, gelen mailleri "spam" veya "spam değil" olarak sınıflandırır. Model, binlerce etiketlenmiş e-postayı inceler ve hangi örüntülerin spam olma olasılığını artırdığını öğrenir. Örneğin "ücretsiz" veya "kazandınız" gibi kelimeler spam ihtimalini yükseltebilir.
- **Regresyon örneği**: Bir emlak sitesi, ev fiyatlarını tahmin etmek için denetimli öğrenme kullanır. Girdiler: evin metrekaresi, oda sayısı, semt. Çıktı: evin fiyatı. Model, geçmiş satış verilerini analiz ederek yeni evlerin fiyatını tahmin eder.

Denetimli öğrenmenin başarısı, veri miktarına ve kalitesine bağlıdır. Yeterince çeşitlilik ve doğruluk yoksa, model hatalı tahminler üretebilir.

Mini Deney

Amaç: Sınıflandırma mantığını anlamak.

- Bir kağıda 10 meyve çizin ve her birini etiketleyin: Elma, Muz, Portakal.
- Bir arkadaşınızın gözlerini kapatın ve sadece özellikleri (rengi, büyüklüğü) söyleyin. Arkadaşınızın hangi meyve olduğunu tahmin etmesini isteyin.

Öğrenilen kavram: Denetimli öğrenmede model, *girdi ve doğru çıktılar* arasındaki ilişkiyi öğrenir.

Ek Örnek: Regresyon

- Evin metrekaresi ve oda sayısı ile tahmini fiyatını hesaplamak için bir tablo oluşturun.
- Geçmiş verilerden örüntüleri çıkarın ve yeni bir evin fiyatını tahmin edin.

Denetimsiz Öğrenme

Denetimsiz öğrenmede makineye yalnızca *girdi verisi* verilir; doğru çıktılar yoktur. Model, verideki örüntüleri kendi başına keşfeder ve gruplamalar oluşturur.

- **Kümeleme örneği**: Bir alışveriş sitesi kullanıcıların davranışlarını analiz etmek ister. Model, benzer davranışlara sahip kullanıcıları gruplar: sık elektronik alışverişi yapanlar, sık kıyafet alanlar gibi. Bu sayede site, kişiye özel reklamlar veya öneriler sunabilir.
- **Anomali tespiti örneği**: Bankalar kredi kartı dolandırıcılığını tespit etmek için denetimsiz öğrenme kullanır. Model, normal harcama davranışlarını öğrenir ve alışılmadık bir işlem gördüğünde uyarı verir. Örneğin bir kullanıcı Türkiye'de alışveriş

yaparken aniden Almanya'da yüksek miktarda bir alışveriş yaparsa, sistem bunu şüpheli olarak işaretler.

Denetimsiz öğrenme, verideki gizli yapıları ve ilişkileri keşfetmek için güçlü bir araçtır.

Mini Deney

Amaç: Kümeleme mantığını anlamak.

- Sınıfta herkesin sevdiği meyveleri listeleyin.
- Benzer tercihleri olan kişileri gruplara ayırın. Örneğin sık elma yiyenler, muz sevenler gibi.

Öğrenilen kavram: Denetimsiz öğrenmede doğru cevap yoktur; makine kendi başına *örüntüleri ve gruplamaları* keşfeder.

Anomali Tespiti Örneği

- Kendi harcama veya puan listelerinizi inceleyin. Normal davranışınızın dışında bir alışkanlığı işaretleyin (örneğin hiç tüketmediğiniz bir ürün). Bu anomali, makine öğrenmesinde alışılmadık davranış tespitine örnektir.

Pekiştirmeli Öğrenme

Pekiştirmeli öğrenmede makine *deneme-yanılma yoluyla öğrenir.* Her eylem, bir ödül veya ceza ile değerlendirilir ve model stratejilerini bu geri bildirimlere göre optimize eder.

- **Oyun örneği**: Satranç veya Go oynayan bir bilgisayar programı, kazanma veya kaybetme sonuçlarını analiz eder. Hangi hamlelerin daha iyi sonuç verdiğini öğrenir ve stratejisini geliştirir.
- **Otonom araç örneği**: Kendi kendine giden bir araba, doğru rotayı seçtiğinde "ödül" alır, yanlış hareket yaptığında "ceza" alır. Zamanla trafik kurallarına ve yol durumuna uygun güvenli sürüş stratejilerini öğrenir.

Pekiştirmeli öğrenme, özellikle çevresel etkileşim gerektiren sistemlerde güçlüdür.

Mini Deney

Amaç: Ödül ve ceza mantığını öğrenmek.

- Basit bir oyun seçin: topu hedefe atmak. Her başarılı atış için bir ödül belirleyin (puan, şeker vs.). Başarısız denemelerde ceza belirleyin (puan kaybı).
- Deneme-yanılma yoluyla stratejinizi geliştirin ve daha fazla başarı elde edin.

Öğrenilen kavram: Pekiştirmeli öğrenme, *deneme-yanılma yoluyla strateji geliştirmeye* dayanır.

Şekil 4. Öğrenme Türleri

Ek Örnek: Otonom Araç Simülasyonu

- Masanızda bir kağıt üzerine bir yol çizin. Küçük bir araba veya topu başlangıç noktasından bitiş noktasına götürün.
- Yanlış yollara saptığında ceza, doğru yollarda ödül verin.
- Zamanla daha hızlı ve doğru hareket edecektir.

Algoritma Nedir?

Makine öğrenmesinin temel yapı taşı *algoritmadır.* Algoritma, belirli bir problemi çözmek için tasarlanmış adımlar dizisidir.

- **Karar ağacı örneği**: Bir öğrencinin sınav notunu tahmin etmek için kullanılabilir. Öğrencinin ders çalışma süresi 2 saatten fazla mı? Evet → iyi not alır. Hayır → düşük not alır. Bu basit karar ağacı, algoritmanın mantığını anlamak için yeterlidir. Daha karmaşık algoritmalar, yüzlerce değişkeni aynı anda analiz edebilir.

Algoritmaların gücü, veriyi nasıl analiz ettikleri ve hangi örüntüleri öğrendikleridir. İyi tasarlanmış bir algoritma, makineye sadece verileri okumakla kalmayıp, onlardan anlamlı sonuçlar çıkarma yeteneği kazandırır.

Mini Deney: Karar Ağacı

1. Öğrencinin sınav başarısını tahmin etmek için bir kağıda tablo hazırlayın: ders çalışma süresi, uyku süresi, sınav notu.

2. Basit bir karar ağacı çizin: “Ders çalışma süresi > 2 saat mi? Evet → iyi not, Hayır → düşük not.”
3. Yeni öğrenciler için tahmin yapın.

Öğrenilen kavram: Algoritmalar veriyi analiz eder ve kararlar üretir. Karmaşık algoritmalar bunu binlerce değişkenle aynı anda yapabilir.

Makine Öğrenmesinin Günlük Hayat Örnekleri

Makine öğrenmesi, soyut bir kavram gibi görünse de, hayatımızda hemen her yerde karşımıza çıkar.

- **Sosyal medya**: Instagram ve TikTok, hangi içeriklere daha fazla ilgi gösterdiğinizi analiz eder ve size benzer içerikler önerir.
- **Müzik ve video platformları**: Spotify veya YouTube, geçmiş seçimlerinize göre çalma listeleri oluşturur.
- **Alışveriş siteleri**: Amazon veya Trendyol, geçmiş satın alımlarınıza bakarak size ürün önerilerinde bulunur.
- **Sağlık**: Makine öğrenmesi, röntgen ve MR görüntülerini analiz ederek hastalık teşhisinde yardımcı olur. Örneğin, akciğer röntgeninde anormal gölgeleri tespit edebilir.
- **Ulaşım**: Google Maps veya Waze, trafik verilerini analiz ederek en hızlı ve güvenli rotayı önerir.

Bu örnekler, makine öğrenmesinin hayatımızda ne kadar geniş bir kullanım alanı olduğunu gösterir. Görünmez ama sürekli çalışan bir zekâ katmanı, günlük kararlarımızı ve deneyimlerimizi şekillendirir.

Bölüm Özeti

Makine öğrenmesi, bilgisayarların veri aracılığıyla öğrenmesini ve deneyimlerden çıkarım yapmasını sağlayan bir yapay zekâ dalıdır. Bu bölümde makinenin öğrenmesi için verinin önemi vurgulanır; veri, makine öğrenmesinin “yakıtı”dır. Veriler farklı türlerde olabilir: sayısal, kategorik, metin, görsel veya zaman serisi. Öğrenme sürecinin başarısı, veri toplama, temizleme ve doğru bir şekilde etiketlemeye bağlıdır. Ayrıca, veri kalitesi ve önyargı (bias) modellerin doğruluğunu ve güvenilirliğini doğrudan etkiler.

Büyük veri kavramı, hacmi, hızı ve çeşitliliği yüksek veri kümelerini ifade eder ve modern makine öğrenmesinin temelini oluşturur. Basit veri örnekleri ve mini uygulamalar, temel kavramları deneyimleyerek öğrenmeyi kolaylaştırır.

Makine öğrenmesinde üç temel yöntem vardır:

- **Denetimli öğrenme:** Model, giriş ve çıkış verileriyle eğitilir ve doğru eşleşmeleri öğrenir.

- **Denetimsiz öğrenme:** Model, etiketlenmemiş verilerden örüntü ve gruplamaları keşfeder.
- **Pekiştirmeli öğrenme:** Model, ödül ve ceza sistemi ile en uygun davranışları öğrenir.

Algoritmalar, veriyi işleyen ve modelin öğrenmesini sağlayan adım adım yönergelerdir. Makine öğrenmesi, günlük yaşamda karşımıza pek çok örnekle çıkar: öneri sistemleri, navigasyon ve trafik tahmini, e-posta filtreleme ve kredi puanı tahminleri gibi uygulamalar, hayatımızı kolaylaştırır ve karar süreçlerini destekler.

Özetle, makine öğrenmesi veriye dayanır, doğru veri ve uygun algoritmalarla öğrenir, ve modern hayatın birçok alanında karşımıza çıkar.

Soru-Cevap

1. **Soru:** Veri neden "yeni dünyanın yakıtı" olarak adlandırılır?
   Cevap: Çünkü makine öğrenmesi sistemleri, çalışmak ve öğrenmek için veriye ihtiyaç duyar; veri ne kadar çok ve kaliteli olursa, model o kadar iyi öğrenir.
2. **Soru:** Veri nedir ve hangi türleri vardır?
   Cevap: Veri, bilgi taşıyan ham gözlem veya ölçümlerdir. Türleri: sayısal, kategorik, metin, görüntü ve zaman serisi gibi çeşitlerdir.
3. **Soru:** Veri toplama, temizleme ve etiketleme neden önemlidir?
   Cevap: Makine öğrenmesi doğru ve güvenilir sonuçlar için temiz, eksiksiz ve doğru etiketlenmiş verilere ihtiyaç duyar; hatalı veriler yanlış öğrenmeye yol açabilir.
4. **Soru:** Büyük veri kavramı neyi ifade eder?
   Cevap: Çok büyük hacimli, hızlı değişen ve çeşitli veri kümelerini ifade eder; klasik yöntemlerle işlenmesi zordur.
5. **Soru:** Veri kalitesi ve önyargı neden makine öğrenmesinde önemlidir?
   Cevap: Düşük kaliteli veya önyargılı veriler, hatalı modeller oluşturur ve adaletsiz kararlar alınmasına sebep olabilir.
6. **Soru:** Basit veri örnekleri ve mini uygulamalar ne amaçla kullanılır?
   Cevap: Makine öğrenmesini öğrenenler için küçük ve anlaşılır veri setleri ile temel kavramları deneyimlemek için kullanılır.
7. **Soru:** Makine öğrenmesinde hangi temel öğrenme yöntemleri vardır?
   Cevap: Denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme temel yöntemlerdir.
8. **Soru:** Denetimli öğrenme nasıl çalışır?
   Cevap: Sisteme giriş ve çıkış verileri (etiketli veri) verilerek modelin ilişkiyi öğrenmesi sağlanır; örneğin e-posta filtreleme.

9. **Soru:** Denetimsiz öğrenme ve pekiştirmeli öğrenme arasındaki fark nedir?
   Cevap: Denetimsiz öğrenmede veri etiketlenmemiştir ve model örüntüleri kendisi bulur; pekiştirmeli öğrenmede model, ödül-ceza sistemi ile doğru kararları öğrenir.

10. **Soru:** Makine öğrenmesi günlük hayatta hangi örneklerde kullanılır?
    Cevap: Örnekler: Netflix öneri sistemi, Spotify çalma listeleri, navigasyon ve trafik tahmini, e-posta spam filtreleme, banka kredi puanı tahmini.

👁 GÖREN VE KONUŞAN MAKİNELER

Yapay zekânın en heyecan verici yönlerinden biri, makinelerin artık sadece hesap yapıp örüntü tanımakla kalmayıp, görme ve konuşma yeteneklerine sahip olmaya başlamasıdır. İnsanlar yıllarca makinelerin sadece sayılar ve mantıkla çalışacağını düşündüler, ancak günümüzde yapay zekâ, fotoğrafları, videoları, sesleri ve metinleri anlamlandırabilecek düzeye gelmiştir. Bu gelişmeler sayesinde makineler, insanlarla daha doğal etkileşim kurabilir ve karmaşık görevleri yerine getirebilir hale gelmiştir.

Bilgisayarlı görü, makinelerin görsel dünyayı algılamasını sağlayan teknolojidir. Örneğin akıllı telefonunuzun yüz tanıma sistemi, sabah uyandığınızda ekran kilidini açarken sizin yüzünüzü tanır. Bu sistem, milyonlarca veriyi analiz ederek yüzünüzün belirli noktalarını tanır ve doğru eşleşmeyi sağlar. Benzer şekilde, sosyal medya platformları fotoğrafları otomatik olarak etiketleyebilir ve arkadaşlarınızı tanıyabilir. Endüstride ise bilgisayarlı görü, üretim hatlarında kalite kontrolünü otomatikleştirir; hatalı ürünleri saniyeler içinde tespit eder ve ayrıştırır. Trafik yönetim sistemlerinde kameralar, araçları algılar, sınıflandırır ve hızlarını takip eder, böylece trafik akışını optimize edebilir.

Doğal dil işleme (Natural Language Processing – NLP), makinelerin insan dilini anlamasına ve üretmesine olanak tanır. Bu teknoloji, çeviri uygulamalarından sohbet botlarına kadar birçok alanda kullanılır. Örneğin, Google Translate, milyonlarca cümle üzerinden eğitilmiş bir model ile dilleri doğru şekilde birbirine çevirir. Siri veya Alexa gibi sesli asistanlar, kullanıcıların söylediklerini anlayarak yanıt verir veya görevleri yerine getirir. E-posta filtreleme sistemleri de NLP kullanır; gelen mailleri spam veya normal olarak sınıflandırır. Bu sistemler, kelime seçimleri, cümle yapısı ve gönderen bilgilerini analiz ederek karar verir. NLP'nin uygulama alanı yalnızca metin ve sesle sınırlı değildir; duygu analizi, metin özetleme ve içerik öneri sistemlerinde de yaygın olarak kullanılır.

Üretken yapay zekâ (Generative AI) ise tamamen yeni içerikler yaratabilen sistemleri kapsar. DALL-E ve Midjourney gibi araçlar, verilen metin açıklamalarına uygun özgün görüntüler üretir. Örneğin "uzayda kahve içen bir robot" dediğinizde, sistem hayal gücünüzle uyumlu, tamamen yeni bir görsel oluşturur. Büyük dil modelleri (LLM) ise metin üretiminde kullanılır; bir hikaye yazabilir, makale özetleyebilir veya bir soruya anlamlı cevap verebilir. Bu yetenekler, eğitim, tasarım, sanat ve bilim gibi alanlarda devrim yaratmaktadır. Örneğin bir öğrenci, "Ortaçağda geçen bir bilim kurgu hikayesi" yazmasını istediğinde, model tarihsel ve kurgu öğelerini birleştirerek özgün bir hikaye ortaya koyar.

Gören ve konuşan makineler, yalnızca teknoloji meraklıları için değil, günlük hayatın içinde de giderek daha görünür hâle gelmektedir. Evimizdeki akıllı cihazlar, araçlarımızdaki sensörler ve

bilgisayarlarımızdaki yazılımlar, bilgisayarlı görü ve NLP sayesinde bizimle etkileşim kurar. Üretken yapay zekâ ise hayal gücümüzü somut ürünlere dönüştürmemizi sağlayarak yaratıcılığımızı destekler. Bu bölümde ele alınan teknolojiler, hem güncel örneklerle hem de kavramsal açıklamalarla, okuyucuya yapay zekânın insan algısına ne kadar yakınlaşabildiğini göstermek için tasarlanmıştır.

Bu nedenle bilgisayarlı görü, doğal dil işleme ve üretken yapay zekâ, yapay zekânın en etkileyici ve pratik alanlarını oluşturur. İnsan beyninden ilham alan bu sistemler, makinelerin dünyayı görmesini, anlamasını ve yeni içerikler üretmesini mümkün kılarak, teknolojiyi sadece araç olmaktan çıkarıp yaratıcı bir iş ortağı hâline getirir. Bu bölüm, yapay zekânın bu üç yönünü örneklerle ve açıklamalarla anlatarak okuyucuyu, makinelerin gözleri ve dilleri aracılığıyla nasıl dünyayı kavradığına dair kapsamlı bir anlayışa ulaştırmayı amaçlamaktadır.

Toplamda, bilgisayarlı görü örnekleri ile nesne tanıma ve yüz algılama, NLP örnekleri ile metin ve konuşma analizi, üretken yapay zekâ örnekleri ile yeni içerik üretimi incelenmiş ve günlük hayatta karşılaşabileceğimiz pratik uygulamalarıyla somutlaştırılmıştır. Bu sayede okuyucu, yalnızca teorik bilgi edinmekle kalmaz, aynı zamanda yapay zekânın yaşamımızı nasıl dönüştürdüğünü daha iyi kavrar.

👁 Bilgisayarlı Görü (Computer Vision)

Bilgisayarlı görü, makinelerin dünyayı “görmesini” sağlayan bir yapay zekâ alanıdır. İnsan gözünün ve beyninin yaptığı işi taklit ederek, fotoğrafları, videoları ve canlı görüntüleri analiz eder, anlamlandırır ve karar verme süreçlerine katkıda bulunur. Bu teknoloji, yalnızca nesneleri tespit etmekle kalmaz, aynı zamanda hareketleri izler, yüzleri tanır, yazıları okur ve hatta medikal görüntülerde hastalıkları teşhis edebilir. Günümüzde bilgisayarlı görü, günlük hayatımızın ayrılmaz bir parçası hâline gelmiş ve hem bireysel hem de endüstriyel uygulamalarda devrim yaratmıştır.

Günlük yaşamdan örneklerle konuyu somutlaştırmak gerekirse, telefonunuzdaki yüz tanıma sistemi ilk akla gelen örneklerden biridir. Her sabah ekran kilidini açmak için yüzünüzü kameraya gösterdiğinizde, cihaz sizin yüzünüzü tanır ve kilidi açar. Bu sistem, milyarlarca örnek üzerinden eğitilmiş algoritmalar sayesinde yüzünüzün belirli noktalarını ve oranlarını tanıyabilir. Bankalardaki ATM’ler de benzer bir şekilde, müşterilerin yüzlerini veya kimlik kartlarını tarayarak işlemlerin güvenli bir biçimde gerçekleştirilmesini sağlar. Sosyal medya platformları ise fotoğrafları otomatik olarak etiketleyebilir, arkadaşlarınızı tanıyabilir ve paylaşımları düzenleyebilir. Örneğin, Facebook veya Instagram, bir etkinlikte çekilen fotoğraflarda kişileri tanıyabilir ve otomatik olarak etiket önerebilir.

Bilgisayarlı görü, temel olarak üç aşamada işler: algılama, tanıma ve takip. Algılama aşaması, görüntüde nesnelerin varlığını belirler. Örneğin, bir trafik kamerası yol üzerindeki araçları algılar. Tanıma aşamasında, algılanan nesneler sınıflandırılır: otomobil, kamyon, bisiklet veya yaya. Takip aşaması ise nesnelerin hareketlerini izler; bu sayede otonom araçlar diğer araçların konumlarını ve hızlarını takip ederek güvenli bir rota çizebilir.

Python ile basit bir örnek üzerinden bilgisayarlı görü mantığını gösterebiliriz. OpenCV kütüphanesi, bu alandaki temel araçlardan biridir.

```
import cv2
```

- **cv2** kütüphanesini import ediyoruz. Bu kütüphane, bilgisayarla görme (computer vision) uygulamaları için en yaygın kullanılan Python kütüphanelerindendir. Görüntü işleme, nesne tespiti, yüz tanıma gibi işlemler için kullanılır.

```
# Görüntüyü yükle
image = cv2.imread('kedi.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

1. **cv2.imread('kedi.jpg')**: Diskten 'kedi.jpg' isimli görüntüyü okur ve bir NumPy dizisi (array) olarak image değişkenine atar.
2. **cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)**: Görüntüyü renkli (BGR) formatından gri tonlamalı (grayscale) formata çevirir.
    - Gri tonlamalı görüntüler, yüz veya kedi gibi nesne tespiti için genellikle daha hızlı ve yeterlidir, çünkü renk bilgisi çoğu algoritmada gerekli değildir.

```
# Haar Cascade ile yüz veya kedi nesnesi algılama
cascade = cv2.CascadeClassifier('haarcascade_frontalcatface.xml')
faces = cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)
```

1. **Haar Cascade**: Önceden eğitilmiş bir nesne algılama algoritmasıdır.
    - 'haarcascade_frontalcatface.xml' dosyası, kedi yüzlerini tespit etmek için eğitilmiş bir Haar Cascade modelidir.
2. **cv2.CascadeClassifier**: Bu sınıf ile XML dosyasını yükleriz ve nesne algılamayı hazır hâle getiririz.
3. **detectMultiScale**: Algoritma gri tonlamalı görüntü üzerinde nesneleri arar.
    - scaleFactor=1.1: Görüntüyü farklı ölçeklerde tararken, bir sonraki ölçekle bir öncekine göre 1.1 oranında büyütür.
    - minNeighbors=5: Bir bölgede nesne olarak kabul edilmesi için en az 5 eşleşme olmalı.
4. Sonuç: faces değişkeni, tespit edilen her nesnenin (x, y, w, h) koordinatlarını içerir.
    - x, y → dikdörtgenin sol üst köşesi
    - w, h → genişlik ve yükseklik

```
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (255,0,0), 2)
```

- faces listesindeki her tespit için bir dikdörtgen çizilir:
    - (x, y) → dikdörtgenin sol üst köşesi
    - (x+w, y+h) → dikdörtgenin sağ alt köşesi
    - (255,0,0) → dikdörtgenin rengi (BGR formatında mavi)
    - 2 → dikdörtgenin kalınlığı

Bu sayede tespit edilen kedi yüzü veya nesneler görsel olarak işaretlenmiş olur.

```
cv2.imshow('Detected Faces', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

1. **cv2.imshow**: Görüntüyü bir pencere içinde gösterir.
    - 'Detected Faces' → pencere başlığı
    - image → gösterilecek görüntü
2. **cv2.waitKey(0)**: Kullanıcı bir tuşa basana kadar pencerenin açık kalmasını sağlar.
3. **cv2.destroyAllWindows()**: Açılmış tüm OpenCV pencerelerini kapatır.

Özet Akış

1. Görüntü yüklenir ve gri tonlamaya çevrilir.
2. Haar Cascade sınıflandırıcısı yüklenir ve görüntüde kedi yüzleri tespit edilir.
3. Tespit edilen yüzlerin etrafına dikdörtgen çizilir.
4. Sonuç ekranda gösterilir ve kullanıcı tuşa basınca pencere kapanır.

Bu basit örnek, bilgisayarın bir görüntüde nesneleri nasıl algıladığını gösterir. Öğrenciler, farklı Haar Cascade modelleri ile insan yüzü, göz veya kedi yüzü gibi nesneleri tanıyabilir.

Bilgisayarlı görü uygulamaları yalnızca nesne tanımakla sınırlı değildir; görüntülerin kalitesini artırabilir, medikal görüntüleri analiz edebilir ve endüstride kalite kontrol süreçlerini otomatikleştirebilir. Örneğin bir üretim hattında sensörler ve kameralar, ürünleri saniyeler içinde tarar ve hatalı ürünleri ayırır. Otomobil fabrikalarında, araç montaj hatlarında her parça kamera tarafından incelenir; eksik veya yanlış monte edilmiş parçalar otomatik olarak tespit edilir.

Trafik yönetim sistemleri de bilgisayarlı görüden yoğun biçimde yararlanır. Kameralar, araçları tanır, hız ihlallerini tespit eder ve trafik ışıklarını akıllıca yönetir. Yapay zekâ, binlerce araç verisini saniyeler içinde analiz ederek, insan operatörlerin asla ulaşamayacağı bir hızda karar verir. Örneğin, yoğun bir kavşakta ışıkların süresini optimize etmek için araçların yoğunluğu ve hareket yönleri analiz edilir.

Görüntü tanıma algoritmaları, derin öğrenme ile büyük bir gelişim göstermiştir. Konvolüsyonel sinir ağları (CNN – Convolutional Neural Networks), görüntüdeki piksel örüntülerini öğrenerek nesneleri yüksek doğrulukla tanıyabilir. Örneğin ImageNet yarışmalarında CNN'ler, insan doğruluk oranını geçerek nesne tanıma konusunda çığır açmıştır.

Python ve Keras kullanarak basit bir CNN modeli oluşturmak, bilgisayarlı görü mantığını somutlaştırır. Aşağıdaki örnek, el yazısı rakamları tanıyan bir model gösterir:

1️⃣ Kütüphaneleri yükleme

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
```

- **Sequential**: Katmanları sıralı bir şekilde eklememizi sağlar, tek yönlü (feed-forward) bir model demektir.
- **Conv2D**: 2 boyutlu konvolüsyon katmanı; görsellerdeki kenar, çizgi ve desenleri öğrenir.
- **MaxPooling2D**: Görüntü boyutunu küçültür ve en önemli özellikleri korur.
- **Flatten**: Çok boyutlu veriyi tek boyuta çevirir; dense (tam bağlantılı) katmana bağlamak için gerekli.
- **Dense**: Tam bağlantılı (fully connected) katman; sınıflandırma yapar.
- **mnist**: El yazısı rakamlardan oluşan hazır veri seti (0–9 arası).
- **to_categorical**: Etiketleri "one-hot" kodlamaya çevirir. Yani örnek: 3 → [0,0,0,1,0,0,0,0,0,0].

2️⃣ Veri setini yükleme ve hazırlama

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

- MNIST veri setini otomatik olarak indirir ve **eğitim (train)** ile **test (test)** olarak ayırır.
- X_train ve X_test: görüntüleri (28x28 boyutunda) içerir.
- y_train ve y_test: karşılık gelen rakam etiketleri (0–9) içerir.

```
X_train = X_train.reshape(-1,28,28,1)/255.0
X_test = X_test.reshape(-1,28,28,1)/255.0
```

- **reshape(-1,28,28,1)**: CNN katmanı için giriş şekli (num_samples, height, width, channels) olmalı. Burada 1 → gri tonlama (tek kanal).
- **/255.0**: Pikselleri 0–1 aralığına normalleştirir; eğitim stabilitesi için önemlidir.

```
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

- Rakam etiketlerini **one-hot** formatına çevirir. Örnek: 5 → [0,0,0,0,0,1,0,0,0,0].

3 CNN Modelini oluşturma

```
model = Sequential([
   Conv2D(32, kernel_size=(3,3), activation='relu', input_shape=(28,28,1)),
   MaxPooling2D(pool_size=(2,2)),
   Flatten(),
   Dense(128, activation='relu'),
   Dense(10, activation='softmax')
])
```

1. **Conv2D(32, (3,3), relu, input_shape=(28,28,1))**
   - 32 filtre (feature map) kullanır, her filtre 3x3 boyutunda.
   - relu: aktivasyon fonksiyonu, negatifleri 0'a çevirir.
   - input_shape=(28,28,1): giriş şekli.
2. **MaxPooling2D(pool_size=(2,2))**
   - Görüntüyü yarıya indirir (28x28 → 14x14).
   - En baskın özellikleri korur, gereksiz detayı atar.
3. **Flatten()**
   - 2D veya 3D çıktıyı tek boyutlu vektöre çevirir, dense katman için gerekli.
4. **Dense(128, relu)**
   - 128 nöronlu tam bağlantılı katman, özellikleri birleştirip öğrenir.
5. **Dense(10, softmax)**
   - Çıkış katmanı: 10 sınıf (0–9).
   - softmax: her sınıfın olasılık dağılımını verir, toplam 1.

4 Modeli derleme ve eğitim

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

- **optimizer='adam'**: ağırlıkları güncellemek için Adam algoritmasını kullanır, hızlı ve güvenilir.
- **loss='categorical_crossentropy'**: çok sınıflı sınıflandırma için kayıp fonksiyonu.
- **metrics=['accuracy']**: eğitim sırasında doğruluk ölçülür.

model.fit(X_train, y_train, epochs=5, batch_size=32)

- Modeli eğitir:
    - **epochs=5** → veri seti 5 kez geçilir.
    - **batch_size=32** → her seferinde 32 örnekle ağırlık güncellenir.

5️⃣ Modeli test etme

accuracy = model.evaluate(X_test, y_test)
print("Test doğruluğu:", accuracy[1])

- evaluate → test setinde kayıp ve doğruluk verir.
- accuracy[1] → doğruluk yüzdesi (accuracy).

💡 Özetle:

Bu kod, MNIST el yazısı rakamlarını tanıyacak basit bir CNN modeli kurar, eğitim verisi ile öğrenir, test verisi ile doğruluğunu ölçer. Conv2D ve MaxPooling2D ile görsel özellikleri yakalar, Dense katmanlarla sınıflandırma yapar.

Bu örnek, bilgisayarlı görü uygulamasının temel mantığını somutlaştırır: makine, piksellerdeki örüntüleri öğrenir ve yeni görüntülerde doğru tahminler yapabilir. Model, her rakamın temel hatlarını ve çizgi örüntülerini tanır ve tanımaya çalıştığı yeni rakamları bu bilgiye dayanarak sınıflandırır.

Bilgisayarlı görü ayrıca medikal görüntüleme alanında hayat kurtarıcı uygulamalara sahiptir. Röntgen, MR ve tomografi görüntüleri, yapay zekâ tarafından analiz edilir; anormallikler tespit edilir ve doktorlara öneriler sunulur. Örneğin göğüs röntgeninde bir tümör veya akciğer iltihabı bilgisayarlı görü sistemleri tarafından hızlıca belirlenebilir. Bu sayede hem teşhis süresi kısalır hem de insan hatası minimize edilir.

Endüstriyel otomasyonda, bilgisayarlı görü kalite kontrolün merkezinde yer alır. Üretim hatlarında milyonlarca ürün, saniyeler içinde taranır ve hatalı ürünler ayrılır. Bu sistemler, insan gözüyle fark edilemeyecek küçük kusurları bile tespit edebilir. Tarım sektöründe, bilgisayarlı görü dronelar kullanılarak ürünlerin büyüme durumu takip edilir ve hastalıklar erken teşhis edilir.

Sonuç olarak, bilgisayarlı görü, makinelerin gözleri olarak görev yapar ve insan yeteneklerini taklit ederek farklı alanlarda kullanılır. Günlük yaşamda yüz tanıma, sosyal medya etiketleme ve trafik yönetimi gibi uygulamalarla karşımıza çıkarken; endüstri, sağlık ve tarım gibi kritik alanlarda ise otomasyon, kalite kontrol ve erken teşhis gibi işlevleri yerine getirir. Konvolüsyonel sinir ağları gibi derin öğrenme teknikleri sayesinde bilgisayarlı görü, giderek daha karmaşık ve hassas görevleri başarıyla yerine getirebilir hale gelmiştir. Python ve Keras gibi araçlar, hem eğitimde hem de uygulamada bilgisayarlı görü konseptlerini somutlaştırmayı sağlar ve öğrencilerin, araştırmacıların veya meraklıların bu teknolojiyi kendi projelerinde deneyimlemelerine olanak tanır.

Bu bölümdeki örnekler, bilgisayarlı görü teknolojisinin temel mantığını, günlük hayat uygulamalarını ve pratik kullanım örneklerini kapsamlı biçimde göstermektedir. Bu sayede okuyucu, makinelerin görme yeteneğini nasıl kazandığını ve bu yeteneğin hayatımıza nasıl entegre olduğunu somut olarak anlayabilir.

Doğal Dil İşleme (Natural Language Processing – NLP)

Doğal Dil İşleme, yani NLP, yapay zekânın en heyecan verici ve hayatımızı doğrudan etkileyen alanlarından biridir. İnsanların günlük hayatlarında fark etmeden kullandıkları pek çok uygulama, NLP'nin gücü sayesinde mümkün olur. NLP, makinelerin insan dilini anlamasını, yorumlamasını ve üretmesini sağlayan bir teknolojidir. Bu teknoloji, çeviri, sohbet botları, metin özetleme, duygu analizi, soru yanıtlama ve hatta metin tabanlı oyunlar gibi birçok alanda kullanılmaktadır.

Günlük yaşamdan örneklerle başlayalım. E-posta uygulamalarındaki spam filtreleri NLP sayesinde çalışır. Gelen kutunuza düşen e-postalar, bir model tarafından "spam" veya "normal" olarak sınıflandırılır. Model, binlerce e-posta üzerinden eğitilmiş ve çeşitli ipuçlarını öğrenmiştir: belirli kelimeler, konu satırları, gönderen adresleri ve hatta e-postanın yapısı. Örneğin, "Buy now!", "Limited offer" gibi ifadeler spam olasılığını artırırken, "Project update" veya "Meeting at 10" gibi ifadeler normal e-posta olarak değerlendirilir. Bu sistem, e-posta kullanıcılarını istenmeyen mesajlardan korur ve aynı zamanda gelen kutusunu düzenli tutar.

Python ile basit bir metin sınıflandırma örneği, NLP'nin temel mantığını anlamak açısından oldukça faydalıdır:

1️⃣ Kütüphaneleri yükleme

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
```

- **CountVectorizer**: Metinleri sayısal forma çevirir. Yani kelime sayımlarını (bag-of-words) yapar.
  Örnek: "Buy now" → [1,1,0,0,...] şeklinde vektör.

- **MultinomialNB**: Naive Bayes sınıflandırıcı, özellikle kelime sayımı gibi **çok terimli olasılık dağılımlarında** iyi çalışır.

2 Eğitim verilerini oluşturma

```
emails = ["Buy now", "Meeting at 10", "Limited offer", "Project update"]
labels = [1,0,1,0]  # 1: spam, 0: normal
```

- emails: metinlerden oluşan eğitim veri seti.
- labels: her e-postanın etiketi.
    - 1 → spam
    - 0 → normal

Örnek:

Email	**Label**
Buy now	1
Meeting at 10	0
Limited offer	1
Project update	0

3 Metinleri sayısal verilere çevirme

```
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(emails)
```

- **fit_transform**:
    1. fit → eğitim verisindeki tüm kelimeleri öğrenir ve kelime sözlüğü oluşturur.
    2. transform → her e-postayı kelime sayısı vektörüne çevirir.

Örnek (sözlük: ["10", "at", "buy", "limited", "meeting", "offer", "project", "update", "now"]):

Email	**Vektör (kelime sayıları)**
Buy now	[0,0,1,0,0,0,0,0,1]
Meeting at 10	[1,1,0,0,1,0,0,0,0]
Limited offer	[0,0,0,1,0,1,0,0,0]
Project update	[0,0,0,0,0,0,1,1,0]

4️⃣ Naive Bayes modeli ile eğitme

```
model = MultinomialNB()
model.fit(X, labels)
```

- MultinomialNB() → Naive Bayes sınıflandırıcısı oluşturur.
- fit(X, labels) → model, hangi kelime kombinasyonlarının spam veya normal e-posta ile ilişkili olduğunu öğrenir.

Naive Bayes, basitçe şöyle çalışır:

$$P(spam \mid email) \propto P(spam) \times \prod P(kelime_i \mid spam)$$

Yani her kelimenin spam olma olasılığı çarpılır ve toplamda spam mi diye tahmin yapılır.

5️⃣ Test e-postası ile tahmin

```
test_email = ["Exclusive offer just for you"]
X_test = vectorizer.transform(test_email)
prediction = model.predict(X_test)
print("Spam mi?", "Evet" if prediction[0]==1 else "Hayır")
```

- vectorizer.transform → test e-postasını aynı kelime sözlüğüne göre sayısal vektöre çevirir.
- model.predict → vektör üzerinde tahmin yapar (1 → spam, 0 → normal).

Örneğin "Exclusive offer just for you" e-postasında offer kelimesi eğitimde spam olarak görülmüş, bu nedenle model muhtemelen **spam** tahmini yapacaktır.

💡 Özet:

1. Metinleri **kelime sayısı vektörüne çeviriyoruz**.
2. **Naive Bayes** ile bu kelime dağılımlarına bakarak sınıflandırma yapıyoruz.
3. Yeni bir e-postayı dönüştürüp predict ile spam mi normal mi olduğunu tahmin ediyoruz.

Bu örnek, bir e-postanın spam olup olmadığını tahmin eder ve öğrencilerin farklı metinler ekleyerek modeli test etmesine olanak tanır. NLP'nin temel mantığı, metinleri sayısal verilere dönüştürmek, örüntüleri öğrenmek ve ardından yeni metinler üzerinde tahminler yapmaktır.

NLP'nin diğer bir önemli uygulama alanı dil çevirisidir. Google Translate, DeepL ve benzeri platformlar, milyarlarca cümle üzerinden eğitilmiş modeller sayesinde farklı diller arasında doğru çeviri yapabilir. Bu sistemler, sadece kelime kelime çeviri yapmaz; bağlamı, dilin yapısını

ve cümlenin mantığını anlamaya çalışır. Örneğin, İngilizce'deki "I'm feeling blue" ifadesi kelime kelime çevrildiğinde anlamsız olur. Ancak gelişmiş bir NLP modeli, bunun "Üzgün hissediyorum" anlamına geldiğini doğru bir şekilde yorumlayabilir.

Sohbet botları ve büyük dil modelleri (Large Language Models – LLM) de NLP'nin günlük hayatımızdaki en görünür uygulamalarındandır. ChatGPT, Siri veya Alexa gibi sistemler, dilin yapısını ve bağlamını öğrenerek kullanıcıların sorularına anlamlı yanıtlar verir. Örneğin bir öğrenci Türkçe olarak bir soru sorduğunda, model bunu anlayabilir ve İngilizce olarak cevaplayabilir. Bu süreç, makinenin yalnızca kelimeleri değil, cümlenin anlamını ve mantığını öğrenmesini gerektirir.

NLP ayrıca metin özetleme ve duygu analizi gibi uygulamalarda da kullanılır. Büyük bir haber makalesini otomatik olarak birkaç cümleye özetleyen sistemler, zaman kazandırır ve kullanıcıya hızlı bilgi sağlar. Duygu analizi ise sosyal medya gönderilerinde veya müşteri yorumlarında kullanılır; sistem, metnin olumlu, olumsuz veya nötr bir duygu taşıyıp taşımadığını belirler. Örneğin bir restoranın yorumlarını analiz eden bir model, genel memnuniyet seviyesini hızlıca ortaya koyabilir.

Python kütüphaneleri bu alanda büyük kolaylık sağlar. Scikit-learn, NLTK, spaCy ve Hugging Face'in Transformers kütüphanesi, NLP modellerini kolayca eğitmek ve test etmek için kullanılabilir. Örneğin Hugging Face Transformers ile metin sınıflandırma veya metin üretimi yapmak mümkündür.

NLP'nin bir diğer önemli yönü de dil modellemedir. Dil modelleri, bir dilin kurallarını ve örüntülerini öğrenir. Bu sayede yeni metinler üretebilir, soruları yanıtlayabilir veya metinler arasında bağlantılar kurabilir. Örneğin bir öğrenci "Matematikte integral nedir?" diye sorabilir ve model, daha önce öğrenmiş olduğu bilgiler doğrultusunda anlaşılır ve doğru bir yanıt üretebilir.

Günlük hayatta NLP'nin farkında olmadan kullanıldığı diğer örnekler de vardır. Akıllı telefonların otomatik yazım düzeltme sistemleri, mesajlaşma uygulamalarındaki öneriler, e-ticaret sitelerindeki ürün açıklamalarının otomatik özetlenmesi ve haber akışlarının kişiselleştirilmesi NLP ile mümkün olmaktadır.

Özetle, Doğal Dil İşleme, insan dilini anlayan, yorumlayan ve üreten makinelerin temelini oluşturur. E-posta filtrelerinden çeviri sistemlerine, sohbet botlarından metin özetleme ve duygu analizine kadar pek çok uygulamada hayatımızı kolaylaştırır. Python ve açık kaynak kütüphaneler sayesinde NLP, hem eğitimde hem de araştırmalarda somut bir şekilde deneyimlenebilir. NLP'nin gücü, yalnızca kelimeleri işlemekle sınırlı kalmayıp, bağlamı anlamak ve anlamlı yanıtlar üretmek gibi insan benzeri yetenekler kazandırmasında yatar.

Bölüm Özeti

Bu bölüm, bilgisayarların görme ve konuşma yeteneklerini nasıl kazandığını açıklar. Bilgisayarlı görü (Computer Vision), bilgisayarların resim ve videoları analiz ederek nesneleri tanımasını ve

anlamasını sağlayan yapay zekâ alanıdır. Günlük hayatta yüz tanıma, otonom araçlar ve tıbbi görüntü analizi gibi uygulamalarda kullanılır.

Doğal Dil İşleme (NLP), bilgisayarların insan dilini anlaması, analiz etmesi ve üretmesiyle ilgilenir. Chatbot'lar, çeviri sistemleri ve sesli asistanlar NLP'nin yaygın kullanım örnekleridir.

Üretken yapay zekâ (Generative AI) ise yeni içerikler üretebilen modelleri kapsar; metin, görüntü, ses veya video oluşturabilir. Örnekler arasında ChatGPT, DALL·E ve müzik üretim sistemleri bulunur. Bu teknolojiler, yaratıcı süreçleri destekler, kişiselleştirilmiş içerik sunar ve günlük yaşamda işleri kolaylaştırır.

Özetle, bu üç alan bir araya geldiğinde makineler görsel ve dil verilerini anlayabilir, yorumlayabilir ve hatta yeni içerik üretebilir; böylece insan benzeri etkileşimler mümkün hâle gelir.

Soru-Cevap

1. **Soru:** Bilgisayarlı görü (Computer Vision) nedir?
   Cevap: Bilgisayarların görsel verileri (resim ve video) analiz edip anlamlandırmasını sağlayan yapay zekâ alanıdır.
2. **Soru:** Bilgisayarlı görü hangi günlük uygulamalarda kullanılır?
   Cevap: Yüz tanıma, otonom araçlar, tıbbi görüntü analizi, güvenlik kameraları ve fotoğraf düzenleme uygulamalarında.
3. **Soru:** Doğal Dil İşleme (NLP) nedir?
   Cevap: Bilgisayarların insan dilini anlamasını, analiz etmesini ve üretmesini sağlayan yapay zekâ dalıdır.
4. **Soru:** NLP hangi uygulamalarda karşımıza çıkar?
   Cevap: Chatbot'lar, dil çeviri sistemleri, metin özetleme, sesli asistanlar ve e-posta filtreleme gibi alanlarda.
5. **Soru:** Üretken yapay zekâ (Generative AI) nedir?
   Cevap: Yeni içerik (metin, görüntü, ses, video) üretebilen yapay zekâ modelleridir.
6. **Soru:** Üretken yapay zekâ örnekleri nelerdir?
   Cevap: ChatGPT, DALL·E, MidJourney, Deepfake ve müzik üretim sistemleri gibi modeller.
7. **Soru:** Bilgisayarlı görü ve NLP arasındaki temel fark nedir?
   Cevap: Bilgisayarlı görü görsel veriyi işlerken, NLP metin veya konuşma gibi dil verilerini işler.
8. **Soru:** Üretken yapay zekâ neden önemli hale gelmiştir?
   Cevap: Yaratıcı içerik üretme kapasitesi, eğitimden sanata kadar birçok alanda yeni fırsatlar sunması nedeniyle.

9. **Soru:** Bilgisayarlı görü ve üretken yapay zekâ birlikte nasıl çalışabilir?
   Cevap: Örneğin bir görüntüye dayalı olarak metin açıklaması üretmek veya fotoğraflardan video oluşturmak gibi görevlerde birlikte kullanılabilirler.

10. **Soru:** Bu teknolojiler günlük yaşamda bizlere ne kazandırır?
    Cevap: İşleri hızlandırır, otomasyonu artırır, kişiselleştirilmiş içerik sunar ve yaratıcı süreçleri destekler.

YAPAY ZEKÂ NERELERDE KULLANILIR?

Yapay zekâ, sadece bilgisayar bilimcilerin laboratuvarlarında kalan soyut bir kavram değildir. Günlük yaşamımızda, farkında olsak da olmasak da her gün yapay zekâ ile etkileşim içindeyiz. Akıllı telefonlarımızın öneri sistemlerinden trafik yönetimine, sağlık taramalarından oyunlara kadar geniş bir yelpazede yapay zekâ, hayatımızı kolaylaştırmak, hızlandırmak ve daha güvenli hâle getirmek için çalışıyor. Bu bölümde, yapay zekânın en öne çıkan kullanım alanlarını örneklerle detaylı şekilde inceleyeceğiz. Sağlık, ulaşım ve otonom sistemler, oyunlar, sanat ve yaratıcılık ile eğitim alanları üzerinden somut örnekler sunacağız.

Sağlık: İnsan Hayatını Kurtaran Algoritmalar

Sağlık sektörü, yapay zekânın en kritik ve etkileyici kullanıldığı alanlardan biridir. Yapay zekâ, hastalıkların erken teşhisinden tedavi planlamasına, hasta takibinden ilaç geliştirmeye kadar pek çok alanda rol oynar. Örneğin radyoloji alanında, bilgisayarlı görü teknikleri kullanılarak röntgen, MR veya CT taramaları incelenir. Model, insan gözünün fark edemeyeceği mikroskobik detayları tespit edebilir. Özellikle kanser gibi kritik hastalıklarda erken teşhis, hayati öneme sahiptir.

Örnek: Bir yapay zekâ sistemi, meme kanseri taramalarında kullanılabilir. Sistem, binlerce örnek taramayı analiz ederek tümörlerin boyutunu ve konumunu işaretler. İnsan doktorlarla birlikte çalıştığında, teşhis doğruluğu ciddi ölçüde artar.

Bir diğer örnek, ilaç keşfi sürecidir. Yeni bir ilaç molekülü tasarlamak yıllar alabilir ve milyonlarca dolara mal olabilir. Yapay zekâ modelleri, kimyasal bileşikleri simüle edebilir ve hangi moleküllerin etkili olacağını tahmin edebilir. Örneğin, protein yapılarının üç boyutlu modellerini tahmin eden algoritmalar, ilaçların hedef proteinlerle etkileşimini önceden gösterebilir. Bu, yeni ilaçların geliştirilme süresini kısaltır ve maliyetleri düşürür.

Ayrıca hasta takibinde yapay zekâ, kronik hastaların verilerini analiz ederek erken uyarılar yapabilir. Örneğin kalp ritmini sürekli izleyen akıllı saatler, anormal bir durum tespit ettiğinde kullanıcıyı ve doktoru uyarabilir. Bu sayede acil müdahale süresi kısalır ve hayat kurtarılır.

Ulaşım ve Otonom Sistemler: Geleceğin Trafiği

Yapay zekâ, ulaşım sektöründe devrim yaratmaktadır. Özellikle otonom araçlar, trafiği daha güvenli ve akıcı hâle getirmeyi hedefler. Bu araçlar, çevrelerini algılamak için bilgisayarlı görü ve sensörleri kullanır. Kamera ve LiDAR verileri, yol, diğer araçlar, yayalar ve engeller hakkında bilgi sağlar. Derin öğrenme algoritmaları, bu verileri analiz ederek aracın hangi hızda ilerleyeceğine, hangi noktada duracağına ve hangi şeritte kalacağına karar verir.

Örnek: Tesla'nın otonom sürüş sistemi, sürekli olarak yol verilerini toplar ve algoritmalarını günceller. Sistem, trafikte beklenmedik bir durumu fark ederek ani fren yapabilir veya şerit değiştirebilir. Yapay zekâ, insan hatalarını minimize ederek kazaları azaltmayı amaçlar.

Toplu taşımada da yapay zekâ etkili bir şekilde kullanılmaktadır. Örneğin, şehir içi otobüs ve metro hatları, yolcu yoğunluğunu tahmin eden algoritmalarla optimize edilir. Trafik ışıkları, yapay zekâ tarafından yönetildiğinde bekleme süreleri azalır ve şehir içi trafik daha akıcı hâle gelir.

Otonom kargo sistemleri de hızla gelişiyor. Örneğin Amazon ve Alibaba gibi şirketler, depo robotları ve insansız teslimat araçları kullanarak lojistik süreçlerini hızlandırıyor. Yapay zekâ, bu araçların güvenli ve verimli çalışmasını sağlamak için gerçek zamanlı kararlar alıyor.

Oyunlar: Yapay Zekâ ile Dinamik Deneyimler

Oyun sektörü, yapay zekânın eğlence dünyasındaki en belirgin örneklerini sunar. Yapay zekâ, oyunlarda düşman karakterlerin davranışlarını belirlemekten, oyuncunun oyun deneyimini kişiselleştirmeye kadar geniş bir kullanım alanına sahiptir.

Örnek: Popüler strateji oyunlarında yapay zekâ, düşman karakterlerin stratejilerini dinamik olarak değiştirir. Oyuncu her hamle yaptığında AI, rakibin hamlelerini tahmin ederek daha zorlu ve gerçekçi bir deneyim sunar. RPG oyunlarında ise yapay zekâ, oyuncunun geçmiş seçimlerini analiz ederek hikâye akışını ve görevleri şekillendirebilir.

Son yıllarda, üretken yapay zekâ oyun tasarımında da kullanılmaya başlandı. Yapay zekâ, yeni haritalar, karakterler ve görevler oluşturabilir. Örneğin, Minecraft gibi sandbox oyunlarında, yapay zekâ rastgele ama mantıklı yapılar yaratabilir ve oyuncuya sonsuz içerik sunabilir.

E-spor alanında ise yapay zekâ, oyuncu performansını analiz ederek taktik önerilerde bulunabilir. Bu, hem profesyonel oyuncuların eğitiminde hem de amatörlerin gelişiminde kullanılabilir.

Sanat ve Yaratıcılık: İnsan Hayal Gücünün Ortağı

Yapay zekâ, yalnızca mantıklı ve veri odaklı alanlarda değil, aynı zamanda sanat ve yaratıcılıkta da etkili bir rol oynamaktadır. Üretken yapay zekâ sistemleri, özgün resimler, müzik parçaları, şiirler ve hatta kısa filmler yaratabilir. İnsan yaratıcılığını destekleyici bir araç olarak kullanılır.

Örnek: DALL-E ve Midjourney, kullanıcıların verdiği metin açıklamalarına uygun görseller yaratır. "Gece gökyüzünde uçan bir kedi" gibi bir komut verildiğinde, sistem daha önce görülmemiş, özgün bir görsel üretir. Bu, sanatçıların ilham almasını ve yeni eserler yaratmasını kolaylaştırır.

Müzik üretiminde yapay zekâ, beste yapmayı kolaylaştırır. Örneğin OpenAI Jukebox veya MusicLM, kullanıcının belirttiği tarz, tempo ve enstrümanlar doğrultusunda yeni parçalar oluşturabilir. Bu parçalar, film müziklerinden video oyunlarına kadar birçok alanda kullanılabilir.

Yapay zekâ aynı zamanda edebiyat ve yazın alanında da kullanılabilir. Büyük dil modelleri, hikâye, şiir veya senaryo yazmak için fikirler üretebilir. Bir yazar, yapay zekâdan ilham alarak kendi eserlerini geliştirebilir veya yazar blokajını aşabilir.

Eğitim: Kişiselleştirilmiş Öğrenme ve Analiz

Eğitimde yapay zekâ, öğrencilere kişiselleştirilmiş öğrenme deneyimleri sunar. Öğrencilerin güçlü ve zayıf yönlerini analiz eden yapay zekâ sistemleri, ders materyallerini ve testleri bireyselleştirir. Bu, öğrencilerin öğrenme sürecini hızlandırır ve etkili hâle getirir.

Örnek: Bir lise matematik öğretmeni, öğrencilerin test sonuçlarını yapay zekâ ile analiz edebilir. Sistem, hangi konularda zorlandıklarını tespit eder ve öğrencilere özel alıştırmalar sunar. Aynı zamanda öğretmenler, yapay zekâ sayesinde hangi konuların genel olarak zor olduğunu görebilir ve ders planlarını buna göre düzenleyebilir.

Online eğitim platformları da yapay zekâdan yararlanır. Örneğin, Khan Academy veya Coursera benzeri sistemlerde, yapay zekâ öğrenciye soru sorabilir, yanlışlarını analiz edebilir ve doğru yanıtları açıklayabilir. Böylece öğrenciler, sınıf ortamına bağımlı kalmadan kendi hızlarında öğrenebilir.

Ayrıca, yapay zekâ öğretmenlerin iş yükünü de azaltır. Otomatik ödev kontrolü, sınav değerlendirmesi ve öğrenci geri bildirimleri gibi rutin işler, yapay zekâ tarafından hızlı ve doğru bir şekilde gerçekleştirilebilir. Bu sayede öğretmenler, öğrencilerle birebir ilgilenmeye daha fazla zaman ayırabilir.

Yapay Zekânın Çapraz Kullanım Örnekleri

Yapay zekâ, çoğu zaman birden fazla alanı aynı anda etkileyebilir. Örneğin:

- **Sağlık ve eğitim:** Tıp öğrencileri için simülasyonlarda hastalık teşhisi yapabilen yapay zekâ modelleri kullanılır.
- **Ulaşım ve oyun:** Otonom araç simülasyonlarında, oyuncular gibi yapay zekâ karakterleri çevreyi analiz eder ve stratejiler geliştirir.
- **Sanat ve eğitim:** Öğrencilere resim veya müzik öğretmek için üretken yapay zekâ modelleri, ders materyali veya yaratıcı egzersizler oluşturabilir.

Bu çapraz kullanım, yapay zekânın esnekliğini ve çok yönlülüğünü gösterir. Tek bir algoritma veya model, farklı bağlamlarda yeni ve anlamlı çıktı üretebilir.

Gelecek Perspektifi

Yapay zekâ, günümüzde zaten hayatımızın birçok alanına nüfuz etmiş olsa da, gelecekte etkisi daha da artacaktır. Sağlık alanında kişiselleştirilmiş tedaviler, ulaşımda tamamen otonom şehirler, eğitimde her öğrenciye özel öğrenme programları, sanat ve oyunlarda sonsuz yaratıcılık imkânları sunulacaktır.

Ancak bu gelişmeler, etik sorular ve güvenlik kaygılarını da beraberinde getirir. Örneğin:

- Sağlık verilerinin gizliliği
- Otonom araç kazaları ve sorumluluk
- Yapay zekânın ürettiği içeriklerin telif hakları
- Eğitimde algoritmik önyargılar

Bu nedenle, yapay zekânın yaygınlaşması ile birlikte etik ve sorumlu kullanım prensipleri geliştirmek büyük önem taşır.

Bölüm Özeti

Bu bölümde yapay zekânın yalnızca teorik bir kavram olmadığı, günlük yaşamın pek çok alanında aktif olarak kullanıldığı vurgulanmıştır. Akıllı telefonlardan sağlık sistemlerine, eğitimden sanata kadar geniş bir kullanım alanına sahip olan yapay zekâ; hayatı kolaylaştırma, hızlandırma ve daha güvenli hâle getirme amacı taşır.

Sağlık alanında, yapay zekâ hastalıkların erken teşhisinde, tedavi planlamasında ve ilaç geliştirme süreçlerinde önemli rol oynar. Özellikle radyoloji görüntülerinin analizinde insan gözünün fark edemeyeceği detayları tespit edebilir. Kanser gibi hastalıklarda erken teşhis oranını artırırken, akıllı cihazlar sayesinde hastaların sürekli izlenmesi de mümkün hâle gelir. Ayrıca ilaç keşfi süreçlerini hızlandırarak maliyetleri düşürür.

Ulaşım ve otonom sistemlerde, yapay zekâ trafik güvenliğini artırmayı ve ulaşımı daha verimli hâle getirmeyi amaçlar. Otonom araçlar sensörler ve kameralar aracılığıyla çevrelerini algılar ve gerçek zamanlı kararlar alır. Bunun yanında trafik ışıklarının yönetimi ve toplu taşıma planlaması gibi alanlarda da yapay zekâ kullanılarak şehir içi ulaşım optimize edilir.

Oyun sektöründe, yapay zekâ daha gerçekçi ve dinamik deneyimler sunar. Oyuncunun davranışlarına göre değişen rakip stratejileri ve kişiselleştirilmiş hikâyeler oluşturulabilir. Ayrıca üretken yapay zekâ sayesinde oyun içerikleri (harita, karakter, görev vb.) otomatik olarak üretilebilir.

Sanat ve yaratıcılık alanında, yapay zekâ insan hayal gücünü destekleyen bir araç hâline gelmiştir. Metin, görsel ve müzik üretimi yapabilen sistemler, sanatçılara ilham verir ve üretim süreçlerini hızlandırır. Aynı zamanda yazarlık ve içerik üretimi gibi alanlarda da yaratıcı destek sunar.

Eğitimde, yapay zekâ kişiselleştirilmiş öğrenme imkânı sağlar. Öğrencilerin güçlü ve zayıf yönlerini analiz ederek onlara özel içerikler sunar. Online eğitim platformlarında öğrencilerin performansını takip eder ve geri bildirim verir. Ayrıca öğretmenlerin iş yükünü azaltarak eğitim sürecini daha verimli hâle getirir.

Bunun yanı sıra yapay zekâ, farklı alanların kesişiminde de kullanılabilir. Sağlık, eğitim, ulaşım ve sanat gibi alanlar arasında köprü kurarak çok yönlü çözümler üretir.

Son olarak, yapay zekânın gelecekte çok daha yaygın ve etkili olacağı öngörülmektedir. Ancak bu gelişmelerle birlikte veri gizliliği, etik sorunlar ve güvenlik riskleri de önem kazanmaktadır. Bu nedenle yapay zekânın bilinçli ve sorumlu kullanımı büyük önem taşır.

Soru-Cevap

1. Yapay zekâ günlük hayatta nerelerde karşımıza çıkar?
Yapay zekâ; akıllı telefonlardaki öneri sistemlerinde, navigasyon uygulamalarında, sağlık hizmetlerinde, online eğitim platformlarında ve oyunlarda sıkça kullanılır.

2. Yapay zekâ sağlık alanında nasıl kullanılır?
Hastalıkların erken teşhisinde, tıbbi görüntülerin analizinde, ilaç geliştirme süreçlerinde ve hasta takibinde kullanılır.

3. Yapay zekâ kanser teşhisinde nasıl yardımcı olur?
Röntgen, MR veya CT görüntülerini analiz ederek tümörleri erken aşamada tespit edebilir ve doktorlara destek olur.

4. Otonom araçlar nasıl çalışır?
Kamera, sensör ve LiDAR gibi teknolojilerle çevrelerini algılar; yapay zekâ algoritmaları sayesinde hız, yön ve fren gibi kararları otomatik olarak verir.

5. Yapay zekâ trafik yönetimini nasıl iyileştirir?
Trafik yoğunluğunu analiz ederek trafik ışıklarını optimize eder ve ulaşımı daha akıcı hâle getirir.

6. Oyunlarda yapay zekâ ne işe yarar?
Oyuncunun hareketlerine göre tepki veren karakterler oluşturur, oyun deneyimini daha gerçekçi ve dinamik hâle getirir.

7. Yapay zekâ sanat alanında nasıl kullanılır?
Metinlerden görseller oluşturabilir, müzik besteleyebilir ve hikâye yazımında yaratıcı fikirler sunabilir.

8. Eğitimde yapay zekâ öğrencilere nasıl yardımcı olur?
Öğrencilerin seviyesine göre içerik sunar, eksik oldukları konuları belirler ve kişiselleştirilmiş öğrenme deneyimi sağlar.

9. Yapay zekâ öğretmenlerin işini nasıl kolaylaştırır?
Ödevleri otomatik olarak değerlendirebilir, öğrenci performansını analiz edebilir ve geri bildirim sağlayabilir.

10. Yapay zekânın gelecekteki etkileri nelerdir?
Daha akıllı şehirler, kişiselleştirilmiş sağlık hizmetleri ve eğitim sistemleri gibi birçok alanda büyük dönüşümler yaratması beklenmektedir; ancak etik ve güvenlik konuları da önem kazanacaktır.

🟦 YAPAY ZEKÂ NASIL ÖĞRENİR?

Yapay zekânın temel yeteneği, öğrenebilmesidir. İnsanlar gibi makineler de deneyimlerinden öğrenebilir. Ancak burada "deneyim" veri ile temsil edilir. Yapay zekâ sistemleri, verileri alır, örüntüleri tanır ve gelecekteki durumları tahmin edebilecek modeller oluşturur. Bu bölümde, bir yapay zekânın nasıl öğrendiğini, eğitim ve test verisinin rolünü, hata ve öğrenme ilişkisini, basit bir modelin adım adım kurulmasını ve aşırı öğrenme (overfitting) kavramını sezgisel olarak inceleyeceğiz.

Eğitim ve Test Verisi

Yapay zekâ modelleri, tıpkı bir öğrencinin sınav öncesi çalışması gibi, önce *veri üzerinden eğitilir.* Eğitim sırasında model, verilerdeki örüntüleri öğrenir ve belirli bir görev için uygun tahminler yapmayı öğrenir. Ancak bir modelin yalnızca öğrendiği verilere bakarak değerlendirilmesi doğru değildir. İnsan gibi, modelin de öğrendiği bilgiyi *bilmediği yeni durumlarda test etmesi gerekir.* İşte burada test verisi devreye girer.

Örnek: Bir makine, el yazısı rakamları tanıyacaksa, önce eğitim verisi olarak binlerce el yazısı rakam görüntüsü ile eğitilir. Ancak gerçek hayatta karşılaşacağı yeni rakamları doğru tanıyabilmesi için ayrı bir test veri seti kullanılır. Eğitim verisi ile test verisinin ayrılması, modelin gerçekten öğrenip öğrenmediğini anlamamızı sağlar.

- *Eğitim verisi (training data):* Modelin öğrenmesi için kullanılan veri.
- *Test verisi (test data):* Modelin doğruluğunu değerlendirmek için kullanılan, daha önce görmediği veri.

Günlük hayattan örnek: Spotify, bir kullanıcının müzik zevkini anlamak için önce geçmiş dinleme verilerini analiz eder (eğitim verisi). Daha sonra, yeni şarkıları önerirken kullanıcının beğenip beğenmediğini gözlemler (test verisi).

Hata ve Öğrenme İlişkisi

Her öğrenme sürecinde hatalar önemlidir. İnsanlar gibi yapay zekâ modelleri de *hatalardan öğrenir.* Modelin çıktısı, gerçek sonuçlardan farklıysa bir hata oluşur ve model bu hatayı azaltmak için ağırlıklarını günceller. Bu süreç, geri yayılım (backpropagation) ve optimizasyon algoritmaları ile gerçekleştirilir.

Sezgisel bir örnek: Bir çocuk ilk kez bisiklete binerken düşer. Düşmek hatadır, ama çocuk düşerek hangi denge hareketlerinin işe yaradığını öğrenir. Yapay zekâ da benzer şekilde, modelin çıktısı ile gerçek sonuç arasındaki farkı ölçer ve bir sonraki tahminde hatayı azaltacak şekilde kendini günceller.

Matematiksel olarak hata, genellikle *kayıp fonksiyonu (loss function)* ile ölçülür. Kayıp fonksiyonu, tahmin edilen değer ile gerçek değer arasındaki farkı sayısal olarak gösterir. Örnekler:

- Ortalama kare hata (Mean Squared Error, MSE): Sürekli değer tahminlerinde kullanılır.
- Çapraz entropi (Cross-Entropy): Sınıflandırma problemlerinde kullanılır.

Python mini örneği:

```
# Basit hata hesaplama örneği
gercek = [1, 0, 1, 1]  # gerçek değerler
tahmin = [0.9, 0.1, 0.8, 0.6]  # modelin tahminleri

mse = sum((g-t)**2 for g,t in zip(gercek,tahmin))/len(gercek)
print("Ortalama Kare Hata:", mse)
```

Bu örnek, modelin tahminlerinin ne kadar doğru olduğunu sayısal olarak görmemizi sağlar. Hata ne kadar küçükse model o kadar iyi öğrenmiştir.

Basit Bir Modelin Adım Adım Kurulması

Bir yapay zekâ modelinin nasıl kurulduğunu adım adım görmek, öğrenme sürecini anlamayı kolaylaştırır. Basit bir sınıflandırma modeli üzerinden gidebiliriz:

1. **Problem Tanımı:** Örneğin bir öğrencinin sınav notunu tahmin etmek veya bir meyvenin elma mı, armut mu olduğunu belirlemek.
2. **Veri Toplama:** Mevcut verileri toplamak. Örneğin boy, ağırlık, renk gibi özellikler.
3. **Veri Hazırlama:** Eksik verileri doldurmak, hatalı değerleri düzeltmek ve gerekirse normalize etmek.
4. **Model Seçimi:** Basit bir perceptron veya lojistik regresyon modeli.
5. **Eğitim:** Model, eğitim verisi üzerinden ağırlıkları öğrenir.
6. **Test:** Modelin performansı daha önce görmediği test verisi ile ölçülür.
7. **İyileştirme:** Hatalar göz önünde bulundurularak model yeniden eğitilir veya parametreler ayarlanır.

Python mini örneği:

```
from sklearn.linear_model import LogisticRegression

# Basit veri seti: boy ve ağırlığa göre elma(0) veya armut(1)
X_train = [[150, 60], [160, 65], [170, 70], [140, 55]]
y_train = [0, 0, 1, 0]
```

```
model = LogisticRegression()
model.fit(X_train, y_train)

# Test verisi
X_test = [[155, 63], [165, 68]]
print("Tahminler:", model.predict(X_test))
```

Bu örnek, modelin eğitim ve test verisi ile nasıl çalıştığını, basit bir sınıflandırma problemini somutlaştırır.

Aşırı Öğrenme (Overfitting) Fikri

Aşırı öğrenme, modelin eğitim verisine çok fazla uyum sağlaması durumudur. Model eğitim verisini neredeyse mükemmel öğrenir, fakat yeni veriler karşısında kötü performans gösterir. İnsan beyninden sezgisel bir örnek verelim: Eğer bir öğrenci sadece geçmiş sınav sorularını ezberler ve mantığı anlamadan çalışırsa, yeni sorularda başarılı olamaz.

Yapay zekâda overfitting, genellikle *çok karmaşık modeller veya çok az veri* ile ortaya çıkar. Basit bir çözüm:

- Daha fazla veri kullanmak
- Modelin karmaşıklığını azaltmak (daha az gizli katman veya nöron)
- Düzenlileştirme (regularization) teknikleri uygulamak

Örnek: El yazısı rakam tanıma modelinde, model yalnızca eğitim setindeki rakamlara uyum sağlarsa, test setindeki farklı yazı tipleri ve stillerde başarısız olur.

Python sezgisel örnek:

1. Gerekli Kütüphanelerin İçe Aktarılması

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
```

Bu bölümde üç önemli araç kullanılıyor:

- **DecisionTreeClassifier** → Karar ağacı modeli oluşturmak için
- **train_test_split** → Veriyi eğitim ve test olarak ayırmak için
- **load_iris** → Ünlü “Iris çiçeği veri setini” yüklemek için

2. Veri Setinin Yüklenmesi

iris = load_iris()

Bu veri seti, üç farklı iris çiçeği türünü sınıflandırmak için kullanılır.
Her örnek için şu bilgiler vardır:

- Taç yaprağı uzunluğu
- Taç yaprağı genişliği
- Çanak yaprağı uzunluğu
- Çanak yaprağı genişliği

Amaç: Bu özelliklere bakarak çiçeğin türünü tahmin etmek.

3. Eğitim ve Test Verisinin Ayrılması

```
X_train, X_test, y_train, y_test = train_test_split(
   iris.data, iris.target, test_size=0.2
)
```

- **X** → Giriş verileri (ölçümler)
- **y** → Çıkış (çiçek türü)

Verinin:

- %80'i **eğitim**
- %20'si **test** için ayrılır

Bu çok önemli çünkü modelin gerçek başarısını görmek için daha önce görmediği veriler gerekir.

4. Modelin Oluşturulması (Kritik Nokta!)

```
model = DecisionTreeClassifier(max_depth=50)
```

Burada model olarak **karar ağacı** seçiliyor.

- max_depth=50 → Ağacın çok derin olmasına izin veriliyor

Bu ne demek?

👉 Model, veriyi **çok detaylı şekilde parçalayabilir**
👉 Neredeyse her örneği ezberleyebilir

Bu da bizi **aşırı öğrenme problemine** götürür.

5. Modelin Eğitilmesi

```
model.fit(X_train, y_train)
```

Model:

- Eğitim verisini alır
- Kurallar öğrenir
- “Eğer şu özellikler varsa bu çiçektir” gibi kararlar üretir

6. Performansın Ölçülmesi

print("Eğitim doğruluğu:", model.score(X_train, y_train))
print("Test doğruluğu:", model.score(X_test, y_test))

Burada iki farklı başarı ölçülür:

✅ **Eğitim Doğruluğu**

- Modelin **ezberlediği veri üzerindeki başarısı**
- Genellikle **çok yüksek çıkar (%100’e yakın)**

❗ **Test Doğruluğu**

- Modelin **yeni veriler üzerindeki başarısı**
- Genellikle daha düşüktür

7. Bu Örnek Ne Anlatıyor?

Bu kodun vermek istediği ana fikir:

👉 **“Model çok iyi öğrendi” ≠ “Model iyi genelleme yapıyor”**

Çok derin bir karar ağacı:

- Eğitim verisini **mükemmel öğrenir**
- Ama yeni verilerde **hata yapar**

Bu örnekte, eğitim doğruluğu çok yüksek olabilir, ama test doğruluğu düşükse model overfit olmuş demektir. Bu, aşırı öğrenmenin temel göstergesidir.

Özetle

- Yapay zekâ, verilerden öğrenir.
- Eğitim verisi öğrenme sürecinin temelini oluşturur, test verisi modelin genelleme yeteneğini ölçer.
- Hatalar, öğrenmenin kritik bir parçasıdır; model, hatalardan öğrenerek kendini geliştirir.
- Basit modellerle öğrenme adım adım gösterilebilir ve kod örnekleri ile somutlaştırılabilir.

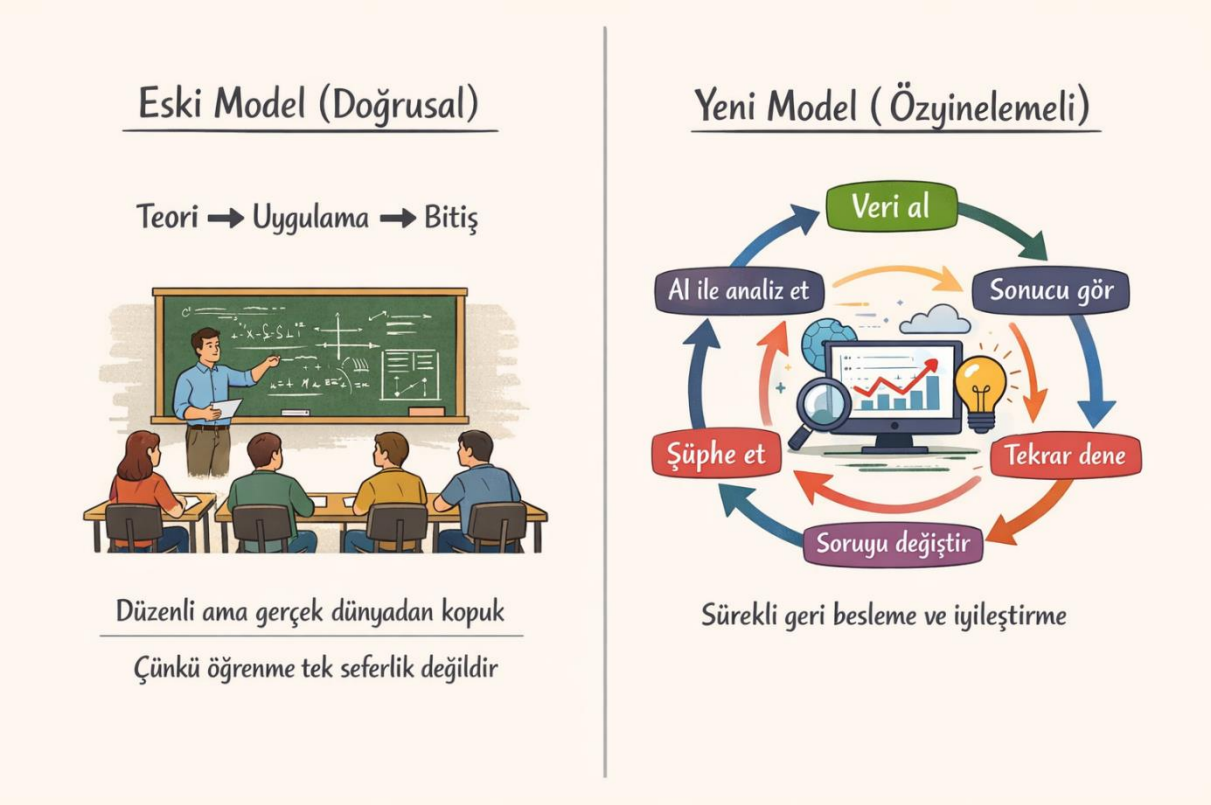
- Overfitting, modelin gerçek hayatta başarısız olmasına yol açar ve dikkatle önlenmelidir.

Günlük hayatta farkında olmadan kullandığımız yapay zekâ uygulamaları, bu temel süreçlerden geçer: spam filtreleri, müzik öneri sistemleri, yüz tanıma, sağlık teşhisi ve otonom araçlar. Tüm bunlar eğitim, test, hata düzeltme ve aşırı öğrenmeden kaçınma süreçleriyle işler.

Özyinelemeli Öğrenme Nedir?

Özyinelemeli öğrenme, yapay zekânın kendi deneyimlerinden ders çıkararak öğrenmesini sağlayan özel bir yöntemdir. Bu yaklaşım, diğer öğrenme türlerinden farklıdır çünkü burada model, doğru cevabı doğrudan gösterilerek öğrenmez; bunun yerine bir ortamla etkileşim kurar, deneme-yanılma yoluyla davranışlarını optimize eder ve ödüller veya cezalar aracılığıyla hangi davranışların iyi olduğunu öğrenir. Basitçe söylemek gerekirse, özyinelemeli öğrenme, “dene, gör, değerlendir ve tekrar dene” döngüsüne dayanır.

Şekil 5. Eski ve Yeni Öğrenme Modelleri



Bu tür öğrenmeyi insan davranışına benzetmek mümkündür. Bir çocuk bisiklet sürmeyi öğrenirken, ilk başta dengesini kaybeder ve düşer. Düşmesi ona bir geri bildirim sağlar: “Bu hareket güvenli değil, başka bir şekilde denemeliyim.” Her denemede çocuğun dengesi ve koordinasyonu artar. Özyinelemeli öğrenme de tam olarak bu şekilde çalışır; model, hatalarını ve doğru hareketlerini gözlemleyerek davranışlarını geliştirir.

Şekil 6. Eski ve Yeni Nesil Öğrenci

Özyinelemeli öğrenmenin temel unsurları dört başlık altında toplanabilir: *agent (ajan), environment (çevre), action (eylem) ve reward (ödül).* Ajan, öğrenen sistemdir. Çevre, ajan ile etkileşimde bulunduğu dünya veya simülasyon ortamıdır. Ajan, belirli eylemleri gerçekleştirir ve çevreden geri bildirim alır. Bu geri bildirim, ödül veya ceza biçimindedir ve ajanın hangi eylemleri daha fazla veya daha az yapması gerektiğini belirler. Örneğin bir oyun oynayan yapay zekâ, puan kazanırsa ödül alır; yanlış bir hamle yaparsa ceza alır.

Bir örnek üzerinden düşünelim. Basit bir labirent oyunu hayal edin. Ajanın hedefi, labirentin çıkışına ulaşmaktır. Başlangıçta ajan rastgele hareket eder ve sık sık duvarlara çarpar. Ancak her başarılı hareket ve doğru yol seçimi için ödül alır. Zamanla ajan, hangi yolların çıkışa daha hızlı ulaştırdığını öğrenir. Bu süreç, özyinelemeli öğrenmenin klasik bir örneğidir.

Özyinelemeli öğrenme, günümüzde birçok alanda uygulanmaktadır. Örneğin otonom araçlar, trafiğin içinde güvenli bir şekilde hareket etmeyi öğrenmek için özyinelemeli öğrenme tekniklerinden faydalanır. Başlangıçta araçlar rastgele hız ve yön değiştirir, çarpışmalardan ve trafik kurallarından ödül/ceza alır ve zamanla güvenli sürüşü öğrenir. Benzer şekilde oyun dünyasında, yapay zekâ ajanları satranç veya Go gibi oyunlarda insanları yenebilecek stratejileri özyinelemeli öğrenme ile keşfeder.

Bir diğer uygulama alanı ise robotik dünyadır. Robotlar, karmaşık görevleri, örneğin bir nesneyi tutmak veya hareket ettirmek gibi işleri, özyinelemeli öğrenme ile öğrenirler. Robot, her denemesinde başarılı veya başarısız eylemleri değerlendirir ve zamanla görevleri hatasız ve verimli bir şekilde yerine getirmeyi öğrenir.

Özyinelemeli öğrenmenin güçlü yönlerinden biri, belirsiz ve değişken ortamlarla başa çıkabilmesidir. Model, her duruma önceden programlanmak yerine kendi deneyimlerinden

öğrenir. Bu, özellikle klasik programlamanın yetersiz kaldığı karmaşık ve dinamik sistemlerde büyük avantaj sağlar.

Ancak özyinelemeli öğrenmenin bazı zorlukları da vardır. Öğrenme süreci genellikle yavaş ve kaynak tüketici olabilir. Ajanın doğru stratejiyi bulması için çok sayıda deneme yapması gerekebilir. Ayrıca ödül sisteminin doğru tasarlanması çok önemlidir; yanlış veya eksik ödül, modelin yanlış öğrenmesine neden olabilir.

Özyinelemeli öğrenme, modern yapay zekâ araştırmalarının merkezinde yer alır. Büyük dil modelleri veya üretken yapay zekâ sistemleri gibi uygulamalar için doğrudan kullanılmasa da, oyun oynayan yapay zekâlar, robotlar ve otonom sistemler için vazgeçilmez bir öğrenme paradigmasıdır.

Sonuç olarak özyinelemeli öğrenme, yapay zekânın çevresiyle etkileşim kurarak, deneme-yanılma yoluyla ve ödüller aracılığıyla öğrenmesini sağlayan güçlü bir yöntemdir. İnsan öğrenmesini taklit eden bu yaklaşım, karmaşık ve belirsiz ortamlarda sistemlerin başarılı olmasını sağlar ve yapay zekâ araştırmalarının en heyecan verici alanlarından biri olarak kabul edilir.

Yapay Zekâ Neden Tekrar Tekrar Dener?

Yapay zekâ sistemlerinin en temel özelliği, deneme-yanılma yoluyla öğrenmeleridir. İnsanlar bir işi ilk yaptıklarında çoğu zaman hatalar yaparlar; tıpkı bir çocuk bisiklete ilk bindiğinde düşmesi gibi. Yapay zekâ da benzer bir süreçten geçer. İlk tahminler genellikle hatalıdır, çünkü model henüz dünyayı ve veri örüntülerini tam olarak anlamamıştır. Bu bölümde, yapay zekânın neden tekrar tekrar denediğini, hataları nasıl ölçtüğünü, öğrenme sürecinde kendini nasıl düzelttiğini ve aynı veriyi tekrar tekrar kullanmanın neden işe yaradığını detaylı şekilde ele alacağız.

İlk Tahmin Neden Hatalı Olur?

Bir yapay zekâ sistemi eğitilmeye başladığında, modelin içindeki ağırlıklar ve bağlantılar genellikle rastgele başlatılır. Bu, tıpkı sıfırdan öğrenen bir öğrencinin ilk cevapları tahmin etmesi gibidir. Örneğin, bir modelin el yazısı rakamları tanıması gerektiğini düşünelim. İlk başta model, “bu rakam 3 mü, 8 mi?” gibi kararlar verirken hatalar yapacaktır. Bunun temel nedeni, modelin henüz veri örüntülerini öğrenmemiş olmasıdır.

İlk tahminlerin hatalı olmasının bir diğer nedeni de modelin karmaşıklığıdır. Basit bir model, verideki tüm nüansları hemen kavrayamaz. Karmaşık bir görüntü tanıma modelinde, piksellerin şekil, gölge ve renk kombinasyonları doğru şekilde yorumlanana kadar sistem yanlış tahminler yapabilir. Örneğin, bir kedi resmi bazen köpek gibi sınıflandırılabilir; çünkü model henüz kedi ve köpek özelliklerini net biçimde öğrenmemiştir.

İlk tahminler hatalı olsa bile bu hatalar, öğrenme sürecinin vazgeçilmez bir parçasıdır. Yapay zekâ, hatalardan ders çıkararak kendini geliştirir. İnsanlar da bir matematik problemi çözerken yanlış cevabı gördüğünde hangi adımı hatalı yaptığını analiz eder ve doğru çözümü öğrenir.

Yapay zekâda da süreç benzer işler; hatalar geri besleme (feedback) mekanizmasıyla modele iletilir.

Hata Nasıl Ölçülür?

Hataları ölçmek için yapay zekâ sistemleri genellikle "kaybı" veya "loss" adı verilen bir ölçüt kullanır. Basit bir sezgisel örnekle anlatmak gerekirse, bir hedefi 10'a vurmak isteyen bir atıcının yaptığı her atışta ne kadar sapma olduğunu düşünün. Eğer hedef 10 ise ve atış 7'ye isabet ettiyse, hata 3 birimdir. Yapay zekâ da benzer şekilde, tahmini ile gerçek sonucu karşılaştırır ve aradaki farkı sayısal bir değer olarak hesaplar.

Örneğin bir e-posta sınıflandırma modeli düşünelim. Bu model, spam ve normal e-postaları ayırmak için eğitiliyor. Eğer model bir spam e-postayı normal olarak sınıflandırırsa, hata oluşur. Hata sayısal bir değer olarak ölçülür ve model bu değeri minimize etmeye çalışır. Başka bir örnek: bir el yazısı rakamları tanıyan model 100 resimden 30'unu yanlış tanırsa, hata oranı %30'dur. Model, hata oranını azaltmak için ağırlıklarını ve bağlantılarını ayarlamaya devam eder.

Model Kendini Nasıl Düzeltir?

Yapay zekânın kendini düzeltme süreci, matematiksel olarak "geri yayılım" (backpropagation) ve "optimizasyon" yöntemleriyle yapılır. Basit bir sezgisel açıklama ile düşünecek olursak, model hatasını gördüğünde adeta kendine şunu sorar: "Bu hatayı nasıl düzeltebilirim?"

Bir örnek üzerinden açıklayalım: bir modelin amacı bir resmi doğru şekilde sınıflandırmaktır. İlk tahmin hatalı olduğunda, model her bir bağlantının ve ağırlığın hataya ne kadar katkıda bulunduğunu hesaplar. Daha sonra bu ağırlıkları hafifçe değiştirerek hatayı azaltmaya çalışır. Bu süreç defalarca tekrar edilir ve zamanla model, doğru tahminler yapmayı öğrenir.

Başka bir örnek: bir dil modeli düşünelim. Model, "Merhaba, nasılsınız?" ifadesini anlamlandırmak için eğitiliyor. İlk başta model yanlış yanıt verebilir. Ancak her yanlış yanıt, modelin parametrelerini ayarlaması için bir fırsattır. Yanlış yanıt sonrası model, hangi kelimelerin ve bağlamların hataya yol açtığını analiz eder ve bir sonraki denemede daha doğru yanıt vermeye çalışır.

Aynı Veriyi Tekrar Tekrar Kullanmak Neden İşe Yarar?

Bir yapay zekâ sistemi, verileri bir kez görmekle sınırlı değildir; veriler defalarca kullanılır. Bunun nedeni, tek bir gösterimle modelin verideki tüm örüntüleri öğrenememesidir. Örneğin, bir çocuk bir kelimeyi yalnızca bir kez duyduğunda öğrenemeyebilir. Ancak kelimeyi tekrar tekrar duydukça ve kullanıldıkça anlam kazanır. Yapay zekâda da süreç benzerdir.

Düşünün ki model, kedileri tanımayı öğreniyor. Model ilk resimde kediyi yanlış tanıyabilir. Ancak aynı veya benzer resimleri tekrar gördüğünde model, hatalarını düzeltir ve örüntüleri daha iyi kavrar. Tekrar tekrar veri görmek, modelin genelleme kapasitesini artırır ve daha doğru tahminler yapmasını sağlar.

Bu süreç, özellikle küçük veri kümeleri veya karmaşık görevlerde çok önemlidir. Büyük veri setleri bile olsa, tekrar eden eğitim döngüleri modelin daha sağlam öğrenmesini sağlar. Ayrıca tekrar eden veri kullanımı, modelin rastgele hatalardan etkilenmesini azaltır ve daha kararlı bir öğrenme sağlar.

"Öğrenme = Düzeltme Süreci" Fikri

Öğrenme sürecinin özü, hataları görmek ve düzeltmektir. Yapay zekâ için öğrenmek, sadece verileri görmek değil, verilerle yapılan tahminlerin doğruluğunu sürekli kontrol etmek ve hataları minimize etmektir. Bu nedenle "öğrenme = düzeltme süreci" ifadesi oldukça doğru bir tanımdır.

Örnek: Bir otonom araç, sensörleri aracılığıyla yol üzerindeki diğer araçları algılar ve hızını ayarlar. İlk sürüşte hatalı bir manevra yapabilir. Bu hata kaydedilir ve simülasyon ortamında veya gerçek sürüşte model, hatayı analiz eder ve bir sonraki denemede daha güvenli bir karar verir. Bu döngü tekrar tekrar devam eder.

Başka bir örnek: bir müzik öneri sistemi düşünelim. Kullanıcının tercihlerini tahmin etmek için öneriler sunar. İlk tahminler yanlış olabilir; kullanıcı bir öneriyi beğenmez. Sistem, bu geri bildirimi hatası olarak değerlendirir ve sonraki önerilerde tercihleri daha iyi tahmin etmeye çalışır. Burada da öğrenme, hataların düzeltilmesine dayanır.

Hata düzeltme süreci, yapay zekânın esnekliğini sağlar. İnsanlar gibi hatalarla karşılaşıp kendini adapte edebilen sistemler, zamanla karmaşık ve değişken görevlerde başarılı olur. Hatalar, modelin öğrenme kapasitesini artıran değerli bilgiler olarak görülür.

Örnek Kod ile Sezgisel Gösterim

Basit bir Python örneği ile hataların nasıl düzeltildiğini görebiliriz. Bu örnek, eğitim ve test verileri ile bir modelin hata üzerinden kendini nasıl geliştirdiğini gösterir:

```
# Basit bir doğrusal regresyon örneği
from sklearn.linear_model import LinearRegression
import numpy as np

# Eğitim verisi
X = np.array([[1],[2],[3],[4],[5]])
y = np.array([2,4,6,8,10])  # Modelin öğrenmesi gereken ilişki: y = 2x

# Model oluştur
model = LinearRegression()

# Modeli eğit
model.fit(X, y)
```

```
# Test verisi ve tahmin
X_test = np.array([[6],[7]])
predictions = model.predict(X_test)

print("Tahminler:", predictions)
```

Bu örnekte model, eğitim verisini kullanarak y = 2x ilişkisini öğrenir. İlk tahminler hatalı olabilir, ancak model tekrar tekrar veri gördükçe hata azalır ve doğruluk artar.

Hata ve Öğrenmenin Hayatımızdaki Karşılığı

Yapay zekânın deneme-yanılma yoluyla öğrenmesi, insan öğrenme süreciyle oldukça benzerdir. Çocuklar yeni bir dil öğrenirken veya bisiklete binerken hata yaparlar. Her hata, bir sonraki denemede daha iyi bir performans için fırsattır. Yapay zekâ da veriler ve hatalar aracılığıyla kendini sürekli geliştirir.

Bir öğrencinin matematik problemlerini çözmesini düşünün. İlk denemede yanlış cevap verebilir. Öğrenci, hatasını analiz eder, öğretmenin geri bildirimini alır ve tekrar dener. Yapay zekâda da süreç aynıdır; model hatasını görür, geri besleme alır ve parametrelerini ayarlayarak bir sonraki tahminde daha doğru olur.

Bölüm Özeti

Yapay zekânın tekrar tekrar denemesi, eğitim ve öğrenme sürecinin temel taşlarından biridir. İlk tahminlerin hatalı olması doğaldır; hatalar geri besleme ile ölçülür ve model bu hataları düzeltir. Aynı verinin tekrar tekrar kullanılması, öğrenmenin sağlam ve kalıcı olmasını sağlar. Öğrenme süreci, hataları görmek ve düzeltmekle eş anlamlıdır. Bu mantık, basit örneklerden karmaşık sistemlere kadar tüm yapay zekâ uygulamalarında geçerlidir.

Yapay zekânın tekrar tekrar denemesi, ona esneklik, adaptasyon ve uzun vadeli başarı kazandırır. İnsan gibi, yapay zekâ da hatalardan ders alır, kendini sürekli geliştirir ve zamanla karmaşık görevlerde bile başarılı olabilir.

Bu süreç, öğrenciler ve yapay zekâ kullanıcıları için de önemli bir ders içerir: hata yapmak normaldir; önemli olan hatalardan öğrenmek ve tekrar denemektir. Yapay zekâ, bu felsefeyi temel alarak, veriyle ve hatalarla büyür ve gelişir.

Soru-Cevap

1. **Soru:** Eğitim ve test verisi nedir ve neden önemlidir?
   Cevap: Eğitim verisi modelin öğrenmesi için kullanılır, test verisi ise modelin ne kadar iyi öğrendiğini değerlendirmek için ayrılır. Bu sayede modelin gerçek hayatta nasıl performans göstereceği anlaşılır.
2. **Soru:** Hata ve öğrenme arasındaki ilişki nedir?
   Cevap: Hata, modelin tahmininin doğruyla ne kadar uyumlu olmadığını gösterir. Model hatalardan öğrenerek performansını artırır.

3. **Soru:** Basit bir model adım adım nasıl kurulur?
   Cevap: Önce veri seçilir, modelin yapısı belirlenir, ağırlıklar rastgele başlatılır ve ardından model hatasını ölçerek ağırlıklar güncellenir.

4. **Soru:** Aşırı öğrenme (overfitting) nedir?
   Cevap: Model, eğitim verisini çok iyi öğrenip genelleme yapamayacak hâle geldiğinde aşırı öğrenme oluşur; test verisinde düşük performans gösterir.

5. **Soru:** Özyinelemeli öğrenme (iterative learning) nedir?
   Cevap: Modelin tekrar tekrar veriyi işleyerek hatalarını düzeltmesi ve performansını artırması sürecidir.

6. **Soru:** Yapay zekâ neden tekrar tekrar dener?
   Cevap: Çünkü tek bir denemede hatasız sonuç almak mümkün değildir; tekrar tekrar deneyerek hata ve doğruluğu ölçer, öğrenir ve kendini iyileştirir.

7. **Soru:** İlk tahmin neden hatalı olur?
   Cevap: Model başlangıçta rastgele ağırlıklarla başlar ve verideki örüntüleri henüz öğrenmemiştir; bu yüzden ilk tahminler genellikle yanlıştır.

8. **Soru:** Hata nasıl ölçülür?
   Cevap: Hata, modelin tahmini ile gerçek değer arasındaki farkla ölçülür; örneğin ortalama kare hata (MSE) veya doğruluk oranı gibi yöntemlerle.

9. **Soru:** Model kendini nasıl düzeltir?
   Cevap: Hataları analiz eder, ağırlıkları günceller ve tekrar deneme yapar; bu süreç, öğrenmenin temel mekanizmasını oluşturur.

10. **Soru:** Aynı veriyi tekrar tekrar kullanmak neden işe yarar?
    Cevap: Tekrar tekrar kullanmak modelin verideki örüntüleri daha iyi öğrenmesini sağlar ve doğruluk oranını artırır. Öğrenme süreci böylece “düzeltme” temeline dayanır.

YAPAY ZEKÂ İLE YAŞAMAK (OKURYAZARLIK)

Yapay zekâ (YZ), artık sadece teknoloji uzmanlarının değil, her bireyin hayatına doğrudan dokunan bir gerçeklik hâline geldi. Telefonlarımız, bilgisayarlarımız, sosyal medya hesaplarımız, hatta evlerimizdeki akıllı cihazlar—tümü yapay zekâ ile çalışıyor. Bu durum, YZ'yi günlük yaşamda nasıl kullanacağımızı bilmenin önemini artırıyor. Yapay zekâ okuryazarlığı, teknolojiyi sadece kullanmak değil, onu doğru ve eleştirel bir bakış açısıyla değerlendirmek anlamına gelir. Bu bölümde, yapay zekâyı bilinçli şekilde kullanmanın yollarını, eleştirel düşünmenin önemini ve YZ ile öğrenme ile YZ'ye bağımlılık arasındaki farkları detaylı örneklerle ele alacağız.

Yapay Zekâyı Doğru Kullanmak

Yapay zekâyı doğru kullanmak, öncelikle teknolojinin sınırlarını ve yeteneklerini anlamakla başlar. Örneğin bir öğrenci, ödev veya proje hazırlarken ChatGPT veya benzeri büyük dil modellerinden yararlanabilir. Burada kritik nokta, yapay zekânın verdiği bilgiyi otomatik olarak doğru kabul etmek değil, onu bir başlangıç noktası olarak kullanmaktır.

Günlük hayattan bir örnek: bir kişi seyahat planı yaparken, yapay zekâ tabanlı uygulamalardan öneriler alabilir. Uygulama "En iyi restoranlar" veya "Gezilecek yerler" gibi listeler sunar. Ancak kullanıcı, yorumları okumalı, güvenilirliğini kontrol etmeli ve kendi tercihlerine göre seçimini yapmalıdır. Bu sayede yapay zekâ, karar verme sürecini destekler ama tamamen yerine geçmez.

Yapay zekânın doğru kullanımında bir diğer önemli unsur, veri gizliliğidir. Örneğin, bir sağlık uygulamasına tüm medikal geçmişinizi yüklemek kolaylık sağlayabilir, ancak verilerin güvenliği ve gizliliği kullanıcı için kritik bir konudur. Yapay zekâyı bilinçli kullanmak, verilerin kimlerle paylaşıldığını ve hangi amaçla işlendiğini anlamayı gerektirir.

Eleştirel Düşünmenin Önemi

Eleştirel düşünme, yapay zekâyı etkin ve güvenli bir şekilde kullanmanın merkezinde yer alır. YZ modelleri, karmaşık algoritmalarla çalışır ve kararları büyük veri örüntülerine dayanarak verir. Bu kararlar çoğu zaman doğru olabilir, ancak modelin hatalı çıktılar üretebileceğini unutmamak gerekir.

Örnek: Sosyal medyada bir haber başlığı yapay zekâ tarafından özetlenmiş olabilir. Kullanıcı, bu özeti okuduğunda başlığın tüm bağlamı yansıtıp yansıtmadığını sorgulamalıdır. Aynı şekilde, e-ticaret sitelerinde önerilen ürünler algoritmaların analizine dayalıdır. Bir ürünün popüler olması, sizin gerçekten ihtiyacınız olduğu anlamına gelmez. Eleştirel düşünce, algoritmanın önerilerini sorgulamak ve kendi mantık süzgecinizden geçirmek anlamına gelir.

Öğrenciler için bir başka örnek: yapay zekâ, ödevler için metin üretebilir veya kod yazabilir. Burada kritik soru şudur: Üretilen içerik gerçekten doğru ve güvenilir mi? Öğrenci, yapay zekânın sağladığı metni veya kodu anlamadan kullanırsa, öğrenme süreci zayıflar. Eleştirel düşünme, kullanıcıyı pasif bir tüketiciden aktif bir değerlendiriciye dönüştürür.

“AI ile Öğrenmek” vs “AI’ye Bağımlı Olmak”

Yapay zekâyı öğrenme sürecinde bir araç olarak kullanmak ile ona bağımlı hâle gelmek arasında ince ama önemli bir fark vardır.

AI ile öğrenmek: Yapay zekâ, öğrencinin anlamasını destekler. Örneğin bir lise öğrencisi, tarih konularını araştırırken yapay zekâdan özetler veya açıklamalar alabilir. Ancak öğrenci bu bilgileri analiz eder, kendi kelimeleriyle tekrar yazar ve tartışırsa öğrenme derinleşir. Yapay zekâ burada bir “yardımcı öğretmen” rolü üstlenir.

AI’ye bağımlı olmak: Öğrenci, yapay zekânın ürettiği tüm bilgiyi sorgulamadan kullanıyorsa, öğrenme süreci zayıflar. Örneğin bir matematik problemi çözülürken öğrenci, yalnızca algoritmanın verdiği sonucu kopyalarsa, problem çözme yeteneği gelişmez. AI’ye bağımlılık, eleştirel düşüncenin devre dışı kalmasına ve bilgiye pasif bir şekilde maruz kalmaya neden olur.

Bu fark, günlük yaşamda da geçerlidir. Örneğin bir kişi, yemek tarifi ararken yapay zekâya danışabilir, ancak tarifi kendi zevkine ve malzemelerine göre uyarlamak öğrenmeyi ve yaratıcı süreci destekler. Sadece algoritmanın önerisini uygulamak, bağımlılığa işaret eder.

Günlük Hayatta Bilinçli Kullanım

Bilinçli kullanım, yapay zekâyı doğru, etik ve güvenli bir şekilde entegre etmeyi içerir. Günlük hayatta bilinçli kullanım örnekleri şunlardır:

1. **Sosyal medya ve haberler**: Algoritmalar, kullanıcılara ilgi alanlarına uygun içerik gösterir. Bilinçli bir kullanıcı, bu filtre balonunun farkında olarak farklı kaynakları da kontrol eder ve doğruluğu teyit eder.
2. **E-posta ve güvenlik**: Yapay zekâ tabanlı spam filtreleri, gelen kutusunu temizler. Ancak bilinçli kullanıcı, önemli mesajların yanlışlıkla silinmediğini kontrol eder.
3. **Alışveriş ve öneri sistemleri**: E-ticaret siteleri, algoritmalarla kişiselleştirilmiş ürün önerileri sunar. Kullanıcı, bu önerilere tamamen güvenmek yerine fiyat ve kalite karşılaştırması yapar.
4. **Sağlık ve fitness uygulamaları**: Yapay zekâ, diyet ve egzersiz önerileri sunabilir. Bilinçli kullanım, önerileri kendi sağlık durumu ve doktor tavsiyeleri ile karşılaştırmayı içerir.
5. **Eğitim ve öğrenme**: Yapay zekâ, öğrencilere kişiselleştirilmiş test ve içerik sağlayabilir. Bilinçli kullanım, öğrencinin YZ'nin sağladığı bilgiyi eleştirel şekilde değerlendirip kendi öğrenme sürecine entegre etmesi demektir.

Bilinçli AI Kullanımı İçin Pratik Öneriler

- **Sorgulayıcı Olun**: YZ'nin verdiği bilgiyi sorgulayın ve doğruluğunu kontrol edin.
- **Bağımsız Öğrenin**: Yapay zekâyı bir destek aracı olarak kullanın, ana bilgi kaynağı olarak değil.
- **Gizliliğe Dikkat Edin**: Kişisel verilerinizi paylaşmadan önce hangi bilgilerin toplandığını ve nasıl kullanılacağını bilin.
- **Çapraz Kaynak Kullanın**: Özellikle eğitim ve haberlerde, farklı kaynaklardan doğrulama yapın.
- **Yaratıcılığı Teşvik Edin**: Yapay zekânın önerilerini kendi fikirlerinizle birleştirerek yenilikçi çözümler üretin.

Bölüm Özeti

Yapay zekâ ile yaşam, sadece teknoloji kullanımı değil, bilinçli düşünme ve etik karar verme sürecidir. Yapay zekâyı doğru kullanmak, eleştirel düşünmek, AI ile öğrenmek ile AI'ye bağımlı olmamak arasındaki farkı anlamak, günlük hayatı bilinçli yönetmek, gizliliği korumak ve etik değerlere dikkat etmek, modern dünyada yaşamanın temel unsurlarıdır.

Telefonlardan sosyal medyaya, eğitimden sağlığa kadar her alanda yapay zekâ ile etkileşim içindeyiz. Ancak farkında olarak ve bilinçli şekilde kullanırsak, YZ hayatımızı destekler ve zenginleştirir. Yanlış veya sorgusuz kullanım ise bağımlılık, yanlış kararlar ve etik sorunlar doğurabilir.

Bilinçli bir kullanıcı, yapay zekâyı hem kendi potansiyelini artırmak hem de çevresine ve topluma olumlu katkı sağlamak için kullanır. Yapay zekâ okuryazarlığı, modern bireyin en temel becerilerinden biri hâline gelmiştir.

Soru-Cevap

1. Yapay zekâ okuryazarlığı nedir?
Yapay zekâ okuryazarlığı, YZ teknolojilerini sadece kullanmak değil; nasıl çalıştığını anlamak, çıktıları sorgulamak ve bilinçli şekilde değerlendirmek anlamına gelir.

2. Yapay zekâyı doğru kullanmak neden önemlidir?
Çünkü YZ her zaman doğru sonuçlar vermez. Bilinçli kullanım, hatalı bilgileri fark etmeyi ve daha doğru kararlar almayı sağlar.

3. Yapay zekâdan alınan bilgiler neden sorgulanmalıdır?
YZ modelleri bazen eksik, yanlış veya bağlamdan kopuk bilgiler üretebilir. Bu nedenle doğruluk kontrolü yapmak gerekir.

4. Veri gizliliği yapay zekâ kullanımında neden önemlidir?
Çünkü birçok YZ sistemi kullanıcı verilerini işler. Kişisel bilgilerin nasıl kullanıldığını bilmek ve korumak güvenlik açısından kritiktir.

5. Eleştirel düşünme yapay zekâ kullanımında ne anlama gelir?
YZ'nin sunduğu bilgileri olduğu gibi kabul etmek yerine analiz etmek, sorgulamak ve farklı kaynaklarla karşılaştırmaktır.

6. "AI ile öğrenmek" ne demektir?
Yapay zekâyı bir yardımcı araç olarak kullanarak bilgiyi anlamak, yorumlamak ve kendi düşüncelerini geliştirmektir.

7. "AI'ye bağımlı olmak" ne demektir?
Yapay zekânın ürettiği bilgileri sorgulamadan kullanmak ve öğrenme sürecini tamamen YZ'ye bırakmaktır.

8. Yapay zekâ öğrencilerin öğrenmesini nasıl destekler?
Kişiselleştirilmiş içerik sunar, eksik konuları belirler ve öğrencilere özel geri bildirim sağlar.

9. Günlük hayatta bilinçli yapay zekâ kullanımına örnekler nelerdir?
Sosyal medyada farklı kaynakları kontrol etmek, alışverişte önerileri karşılaştırmak, sağlık uygulamalarını dikkatli kullanmak ve bilgileri doğrulamaktır.

10. Yapay zekâyı bilinçli kullanmak için hangi temel alışkanlıklar geliştirilmelidir?
Sorgulayıcı olmak, farklı kaynaklardan doğrulama yapmak, veri gizliliğine dikkat etmek, bağımsız düşünmek

ETİK, RİSKLER VE GÜVENLİK

Yapay zekâ artık sadece teknolojik bir araç değil, hayatımızın birçok alanını doğrudan etkileyen bir güç hâline gelmiştir. Bu güç, fırsatları beraberinde getirirken ciddi etik ve güvenlik sorumluluklarını da gündeme taşır. Yapay zekâ algoritmalarının, verilerin ve sistemlerin tasarımı sırasında alınan kararlar, yalnızca teknik doğruluk değil, aynı zamanda toplumsal, hukuki ve ahlaki sonuçlar da doğurur. Bu nedenle yapay zekâ okuryazarlığının bir parçası olarak etik ve güvenlik boyutunu anlamak, günümüz ve geleceğin temel bir gerekliliğidir. Bu bölümde algoritmik önyargı, veri gizliliği, deepfake ve dezenformasyon, yapay zekâ güvenliği ve "Makine karar vermeli mi?" sorusunu detaylı örneklerle ele alacağız.

Algoritmik Önyargı

Algoritmik önyargı, yapay zekâ sistemlerinin kararlarını etkileyen ve çoğunlukla veri kaynaklı olan sistematik hatalardır. Algoritmalar, kendilerine verilen veriyi temel alarak öğrenir ve tahminlerde bulunur. Eğer veride önyargılar varsa, sonuçlar da önyargılı olur. Örneğin, bir işe alım algoritması geçmiş işe alım verilerini analiz ediyorsa ve geçmişte belirli bir cinsiyeti veya etnik grubu tercih ediyorsa, algoritma da bu tercihleri öğrenir ve uygular. Sonuç olarak, yeni başvuru sahipleri adil bir şekilde değerlendirilmez.

Bir başka örnek, yüz tanıma sistemleridir. Bazı araştırmalar, bu sistemlerin açık tenli insanları daha doğru tanırken, koyu tenli bireylerde hata oranının yükseldiğini göstermiştir. Bu durum,

veri setlerinin dengesizliğinden kaynaklanır. Yapay zekâ tasarımcıları, algoritmanın doğruluğunu artırmak ve önyargıyı azaltmak için veriyi çeşitlendirmeli ve model performansını farklı gruplarda test etmelidir.

Algoritmik önyargıyı anlamak, yalnızca teknik bir sorun değil, aynı zamanda toplumsal bir sorumluluktur. Yapay zekâ kararlarının etkileri bireylerin yaşamını doğrudan etkileyebilir; iş bulma, kredi alma, eğitim ve hatta hukuki süreçlerde bu algoritmalar kritik rol oynayabilir. Bu nedenle etik ilkeler ve şeffaflık, algoritma tasarımında en az doğruluk kadar önemlidir.

Veri Gizliliği

Yapay zekânın çalışabilmesi için büyük miktarda veriye ihtiyaç vardır. Bu veriler çoğunlukla kişisel bilgi içerir ve yanlış kullanıldığında ciddi gizlilik sorunlarına yol açabilir. Örneğin, bir sağlık uygulaması hastaların tıbbi geçmişini analiz ederek öneriler sunabilir. Eğer bu veriler kötü niyetli kişilerin eline geçerse, bireylerin mahremiyeti ihlal edilmiş olur.

Gizlilik, yalnızca veri toplamakla ilgili değildir; aynı zamanda veri depolama, iletme ve analiz süreçlerini de kapsar. GDPR (Genel Veri Koruma Yönetmeliği) ve KVKK gibi düzenlemeler, kullanıcı verilerinin nasıl toplanacağını ve işleneceğini belirler. Bir kullanıcı, yapay zekâ tabanlı bir uygulama kullanırken, hangi verilerin toplandığını, kimlerle paylaşıldığını ve ne amaçla kullanıldığını bilmelidir.

Örnek olarak, bir sosyal medya platformu kullanıcı verilerini reklam hedeflemesi için kullanabilir. Bilinçli bir kullanıcı, gizlilik ayarlarını kontrol eder ve kişisel verilerini korumak için gerekli önlemleri alır. Yapay zekâyı bilinçli ve etik şekilde kullanmak, yalnızca teknolojik bilgiye değil, aynı zamanda veri hakları ve gizlilik bilincine de sahip olmayı gerektirir.

Deepfake ve Dezenformasyon

Deepfake teknolojisi, yapay zekâ ile gerçekçi sahte video ve ses içerikleri üretmeyi mümkün kılar. Bu teknoloji, eğlence ve film endüstrisinde yaratıcı bir araç olarak kullanılabilirken, yanlış ellerde ciddi sorunlar yaratabilir. Örneğin, bir politikacının sahte bir videosu, toplumsal algıyı manipüle etmek veya seçimleri etkilemek amacıyla kullanılabilir.

Dezenformasyon, yani yanlış veya yanıltıcı bilginin yayılması, sosyal medya ve dijital platformlarda yaygın bir problemdir. Yapay zekâ bu süreci hem hızlandırabilir hem de etkisini artırabilir. Örneğin, sahte haberleri otomatik olarak üreten ve dağıtan bir algoritma, milyonlarca kişiye kısa sürede ulaşabilir. Bu nedenle, yapay zekâ okuryazarlığı, kullanıcıların içerikleri eleştirel bir gözle değerlendirmesini ve doğruluğunu sorgulamasını gerektirir.

Deepfake ve dezenformasyonla mücadele, yalnızca teknik çözümlerle değil, toplumsal farkındalık ve eğitimle de mümkündür. Kullanıcılar, görsel veya işitsel içeriklerin kaynağını sorgulamalı ve doğrulama araçlarını kullanmalıdır. Bu yaklaşım, yapay zekâ ile yaşarken etik ve güvenli kalmanın temel adımlarından biridir.

Yapay Zekâ Güvenliği

Yapay zekâ güvenliği, sistemlerin hem dış tehditlere karşı korunmasını hem de hatalı kararlar üretmesini önlemeyi kapsar. YZ sistemleri, siber saldırılara açık olabilir ve kötü niyetli kişiler tarafından manipüle edilebilir. Örneğin, otonom araçların kullandığı görsel algılama sistemleri, sahte trafik işaretleri ile kandırılabilir ve kazalara yol açabilir.

Güvenlik yalnızca fiziksel veya dijital tehditlerle sınırlı değildir; aynı zamanda algoritmanın kendisinden kaynaklanan riskleri de içerir. Hatalı eğitim verileri, modelin yanlış kararlar üretmesine sebep olabilir. Örneğin bir kredi değerlendirme algoritması, geçmiş verilerden kaynaklı hatalar nedeniyle düşük gelirli gruplara haksız olarak olumsuz karar verebilir. Bu durum, hem bireyler hem de kurumlar için risk oluşturur.

YZ güvenliği, modelin sürekli izlenmesi, test edilmesi ve hataların hızlı şekilde düzeltilmesi ile sağlanır. Ayrıca etik ilkeler ve şeffaflık, güvenliği artıran önemli unsurlardır. Kullanıcılar ve geliştiriciler, yapay zekânın potansiyel risklerini anlamalı ve gerekli önlemleri almalıdır.

“Makine Karar Vermeli mi?” Sorusu

Yapay zekâ sistemleri, karar verme süreçlerinde insanları destekleyebilir; ancak kritik bir soru ortaya çıkar: Makine karar vermeli mi?

Örneğin, sağlık alanında bir yapay zekâ, hastalık teşhisi ve tedavi önerileri sunabilir. Burada karar tamamen makineye bırakılırsa, etik ve hukuki sorular gündeme gelir. Yanlış teşhis veya hatalı tedavi önerisi, hastanın hayatını tehlikeye atabilir. Bu nedenle, YZ çoğu zaman insan kararını destekleyici bir araç olarak kullanılmalıdır.

Adalet sistemi de benzer bir örnek sunar. YZ tabanlı bir risk değerlendirme algoritması, mahkûmların tekrar suç işleme olasılığını tahmin edebilir. Ancak bu tahminler tamamen makineye bırakılırsa, önyargılar ve hatalı veri nedeniyle adaletsizlik oluşabilir. İnsan gözetimi, etik ve hukuki sorumluluğun temelini oluşturur.

Bu durum, otomasyonun sınırlılıklarını ve insan sorumluluğunu hatırlatır. Yapay zekâ kararları hızlandırabilir, analiz kapasitesini artırabilir ve veri örüntülerini ortaya çıkarabilir; ancak kritik ve etik açıdan hassas kararlar, insan gözetimi ve değerlendirmesi ile dengelenmelidir.

Etik ve Güvenlik İçin Örnek Senaryolar

Bir örnek senaryo üzerinden konuyu somutlaştıralım: Bir şehir, trafikteki kazaları azaltmak için yapay zekâ tabanlı otonom araçlar kullanıyor.

- Algoritmik önyargı: Araç, belirli yaya gruplarının davranışlarını yanlış tahmin edebilir. Önyargılı veri setleri, bazı bölgelerde güvenliği azaltabilir.
- Veri gizliliği: Araçlar, şehirdeki tüm hareketleri kaydediyor. Bu veriler kötü niyetli kişilerin eline geçerse mahremiyet ihlali oluşabilir.
- Deepfake ve dezenformasyon: Sahte trafik videoları, yanlış ihbarlara veya panik yaratmaya neden olabilir.

- Güvenlik: Yazılım açıkları, siber saldırılara ve sistem çökmesine yol açabilir.
- Karar mekanizması: Acil bir durumda aracın ne yapacağı, algoritmanın mı yoksa sürücünün mi karar vereceği etik bir sorundur.

Bu örnek, yapay zekâ uygulamalarının tüm boyutlarda etik ve güvenlik sorumluluğu gerektirdiğini gösterir. Her adımda insan gözetimi, etik değerlendirme ve güvenlik önlemleri kritik öneme sahiptir.

Yapay zekâ etik ve güvenlik boyutları, teknoloji geliştikçe daha da önem kazanacaktır. Makine öğrenmesi ve derin öğrenme algoritmaları daha karmaşık hale gelirken, algoritmik önyargı, veri gizliliği ve güvenlik açıkları daha sofistike tehditler oluşturabilir. Toplumun ve bireylerin bilinçli olması, yalnızca teknolojiyi doğru kullanmak için değil, aynı zamanda toplumsal adalet, güvenlik ve etik değerlerin korunması için de gereklidir.

Eğitim kurumları, işletmeler ve devletler, yapay zekânın etik ve güvenli kullanımını teşvik eden politikalar geliştirmeli, kullanıcıları bilinçlendirmeli ve teknolojiyi sadece işlevsel değil, etik olarak da sürdürülebilir hâle getirmelidir.

Bölüm Özeti

Yapay zekâ, insan hayatını dönüştüren güçlü bir araçtır. Ancak bu güç, etik ve güvenlik sorumluluklarıyla birlikte gelir. Algoritmik önyargı, veri gizliliği, deepfake ve dezenformasyon, yapay zekâ güvenliği ve makinelerin karar verme kapasitesi, sadece teknik değil, toplumsal bir mesele olarak ele alınmalıdır. Kullanıcılar ve geliştiriciler, yapay zekâyı bilinçli, etik ve güvenli bir şekilde tasarlamalı ve kullanmalıdır. Bu sayede yapay zekâ, yalnızca hayatı kolaylaştıran bir araç değil, aynı zamanda güvenli ve adil bir teknoloji olarak toplumda yerini alabilir.

Etik ve güvenlik, yapay zekâyı anlamak ve kullanmak isteyen herkes için temel bir rehberdir. Yapay zekâ okuryazarlığı, teknik bilgi ile etik bilinç arasında bir köprü kurar ve bireylerin hem kendi hayatlarını hem de toplumu güvenli ve adil bir şekilde yönlendirmelerine olanak tanır.

Soru-Cevap

1. Yapay zekâ neden etik ve güvenlik açısından önemlidir?

Çünkü yapay zekâ sistemleri insanların hayatını doğrudan etkileyen kararlar alabilir. Bu kararlar yalnızca teknik değil, aynı zamanda toplumsal ve ahlaki sonuçlar doğurur.

2. Algoritmik önyargı nedir?

Algoritmik önyargı, yapay zekâ sistemlerinin eğitim verilerindeki dengesizlikler nedeniyle adaletsiz veya taraflı sonuçlar üretmesidir.

3. Algoritmik önyargıya günlük hayattan bir örnek nedir?
Bir işe alım algoritmasının geçmişte belirli bir grubu tercih eden verilerle eğitilmesi ve yeni adayları da bu doğrultuda değerlendirmesi buna örnektir.

4. Veri gizliliği neden yapay zekâda kritik bir konudur?
Çünkü yapay zekâ sistemleri büyük miktarda kişisel veri kullanır. Bu verilerin kötüye kullanılması mahremiyet ihlallerine yol açabilir.

5. Kullanıcılar veri gizliliğini korumak için ne yapmalıdır?
Hangi verilerin toplandığını öğrenmeli, gizlilik ayarlarını kontrol etmeli ve gereksiz kişisel bilgileri paylaşmaktan kaçınmalıdır.

6. Deepfake nedir ve neden tehlikelidir?
Deepfake, yapay zekâ ile üretilen sahte video veya ses içerikleridir. Yanlış bilgi yaymak ve insanları yanıltmak için kullanılabilir.

7. Dezenformasyon yapay zekâ ile nasıl yayılabilir?
YZ sistemleri sahte haberleri hızlı şekilde üretip geniş kitlelere ulaştırabilir, bu da yanlış bilgilerin yayılmasını kolaylaştırır.

8. Yapay zekâ güvenliği neyi kapsar?
Sistemlerin siber saldırılardan korunması, hatalı kararların önlenmesi ve algoritmaların güvenilir şekilde çalışmasını sağlar.

9. Yapay zekâ sistemleri tamamen kendi başına karar vermeli midir?
Hayır, özellikle kritik alanlarda (sağlık, hukuk gibi) yapay zekâ kararları insan gözetimi ile desteklenmelidir.

10. Yapay zekâ kullanımında etik ve güvenliği sağlamak için neler yapılmalıdır?
Şeffaflık sağlanmalı, önyargılar azaltılmalı, veri gizliliği korunmalı ve sistemler düzenli olarak test edilmelidir.

YAPAY SİNİR AĞLARI VE DERİN ÖĞRENME

Yapay zekâ dünyasında *yapay sinir ağları (YSA)* ve *derin öğrenme,* bilgisayarların öğrenmesini ve akıllı kararlar vermesini sağlayan en etkili yöntemler arasında yer alır. Günlük hayatta kullandığımız akıllı telefonlar, sosyal medya uygulamaları, hatta otomatik fotoğraf etiketleme sistemleri, çoğunlukla bu teknolojiler sayesinde çalışır. Peki, yapay sinir ağları nasıl öğrenir ve insan beyninden ilham alarak bu kadar karmaşık görevleri yerine getirebilir?

İnsan beyninde yaklaşık 86 milyar nöron bulunur. Nöronlar, birbirleriyle *sinaps* adı verilen bağlantılar üzerinden iletişim kurar. Bu bağlantılar sayesinde beynimiz yeni bilgileri öğrenir, hatırlamaları işler ve kararlar alır. Örneğin, bir el yazısı rakamı gördüğümüzde beynimiz, geçmiş deneyimlerimizle karşılaştırarak bu rakamın “3” mü yoksa “8” mi olduğunu hemen fark eder. Bu karmaşık işlem saniyeler içinde gerçekleşir ve biz çoğu zaman farkına bile varmayız. Yapay sinir ağları, işte bu temel mantığı bilgisayar ortamına taşır.

Yapay sinir ağları, beynin nöronlarına benzer şekilde *küçük bilgi işleme birimlerinden* oluşur. Bu birimlere “yapay sinir hücreleri” veya *perceptron* denir. Her perceptron, aldığı bilgiyi işler ve belirli bir karar verir. Bir perceptron basit bir şekilde çalışır: aldığı veriyi değerlendirir, bir eşik

değerine göre karar verir ve sonucu diğer birimlere iletir. Birden fazla perceptron bir araya geldiğinde, daha karmaşık hesaplamalar ve öğrenme süreçleri mümkün hâle gelir. Bu sayede bilgisayarlar, yalnızca tek bir perceptron ile çözemeyeceği sorunları katmanlı yapılar sayesinde öğrenebilir.

Yapay sinir ağları, genellikle üç temel katmandan oluşur: *giriş katmanı, gizli katmanlar ve çıkış katmanı.* Giriş katmanı, bilgisayara aktarılan bilgiyi alır. Gizli katmanlar, bu bilgiyi işler ve verinin içindeki karmaşık desenleri tanımaya çalışır. Çıkış katmanı ise, ağın öğrendiği bilgiyi bir karar veya tahmin hâline getirir. Örneğin, el yazısı rakam tanıma uygulamasında giriş katmanı pikselleri alırken, gizli katmanlar rakamın kenarlarını, şekillerini ve çizgilerini tanır. Çıkış katmanı ise hangi rakam olduğunu belirtir.

Bu sistemin en büyük avantajlarından biri, *deneme-yanılma yoluyla öğrenebilmesidir.* Bir ağ, yanlış tahminler yaptığında hatasını fark eder ve ağırlıklarını buna göre yeniden düzenler. İnsan beyninin deneyimlerden öğrenmesine çok benzeyen bu mekanizma, bilgisayarların zamanla daha doğru tahminler yapmasını sağlar. Üstelik bu süreç, çok daha hızlı ve büyük miktarda veri ile gerçekleştirilebilir.

Özetle, yapay sinir ağları ve derin öğrenme, insan beyninin işleyişinden ilham alarak geliştirilmiş yöntemlerdir. Küçük bilgi işleme birimleri olan perceptronlar, bir araya gelerek karmaşık bilgileri öğrenir ve kararlar alabilir. Bu sayede bilgisayarlar, yüzleri tanıyabilir, sesleri anlayabilir, hatta oyun oynarken strateji geliştirebilir. Bu bölümde, önce basit perceptronların mantığını öğrenecek, ardından katmanlı yapılar ve derin öğrenmenin temel ilkelerini keşfedeceğiz. Böylece, yapay zekânın karmaşık ama büyüleyici dünyasına ilk adımımızı atmış olacağız.

Yapay Sinir Hücresi (Perceptron)

Yapay sinir ağlarının en temel yapı taşlarından biri perceptrondur. Bir perceptron, bilgisayar ortamında insan nöronlarını taklit eden basit bir bilgi işleme birimidir. Bu bölümde, perceptronun ne olduğunu, nasıl çalıştığını, biyolojik nöronlarla olan benzerliklerini ve basit bir örnek üzerinden nasıl öğrenebildiğini öğreneceğiz.

Perceptron, insan beynindeki nöronlardan ilham alınarak geliştirilmiştir. Beynimizdeki nöronlar, *dendritler, aksonlar ve sinapslar* ile birbirine bağlıdır. Dendritler, diğer nöronlardan gelen sinyalleri alır ve bunları toplar. Eğer bu toplam belirli bir eşik değeri aşarsa, nöron aksonu aracılığıyla diğer nöronlara sinyal gönderir.

Yapay perceptron da aynı mantığı takip eder:

- **Dendrit → Girişler:** Perceptron, dış dünyadan gelen bilgileri girişler aracılığıyla alır.
- **Sinaptik ağırlık → Perceptron ağırlıkları:** Bazı girişler daha önemli olduğunda, bu önem perceptron ağırlıklarıyla ifade edilir.
- **Eşik değeri → Bias:** Perceptron, toplam giriş değerini belirli bir eşik ile karşılaştırır.

- **Akson → Çıkış:** Eğer eşik aşılırsa, perceptron çıktı üretir ve bu çıktı diğer nöronlara veya sistemlere iletilir.

Bu benzerlik, yapay sinir ağlarının neden "beyinden esinlenmiş" olarak adlandırıldığını gösterir. İnsan beyninin bilgi işleme mantığını basit bir şekilde bilgisayar ortamına taşımak, perceptronun temel amacıdır.

Perceptron, bir bilgisayar modelinde karar almayı sağlayan en basit yapay nörondur. Basitçe anlatmak gerekirse, perceptron şu adımları izler:

1. **Girişleri alır:** Perceptron, dış dünyadan gelen bilgileri alır. Bu bilgiler, sayısal değerler veya 0 ve 1 gibi mantıksal değerler olabilir. Örneğin, bir sınıfta öğrencilerin ödev notlarını düşünelim. Her öğrencinin notu, perceptrona bir "giriş" olarak aktarılır.
2. **Ağırlıklarla çarpar:** Her girişin bir önemi vardır. Bazı öğrencilerin notları, diğerlerinden daha etkili olabilir. Bu önemi belirlemek için perceptron, her giriş değerini bir *ağırlık* ile çarpar. Matematiksel olarak, giriş ile ağırlık çarpılır ve bu çarpımlar toplanır.
3. **Toplamı eşik değeriyle karşılaştırır:** Tüm girişlerin ağırlıklı toplamı alındıktan sonra, perceptron bu toplamı bir *eşik değeri (threshold)* ile karşılaştırır. Eğer toplam eşik değerini aşarsa, perceptron "evet" veya "1" gibi bir çıktı üretir; aşmazsa "hayır" veya "0" üretir.
4. **Çıktı üretir:** Sonuç olarak perceptron, aldığı veriyi işleyerek bir çıktı verir. Bu çıktı, bilgisayarın karar verme sürecinin temel adımıdır. Örneğin, bir öğrencinin sınıfı geçip geçmediğini belirleyen bir sistemde, perceptron "geçti" veya "kaldı" şeklinde karar verebilir.

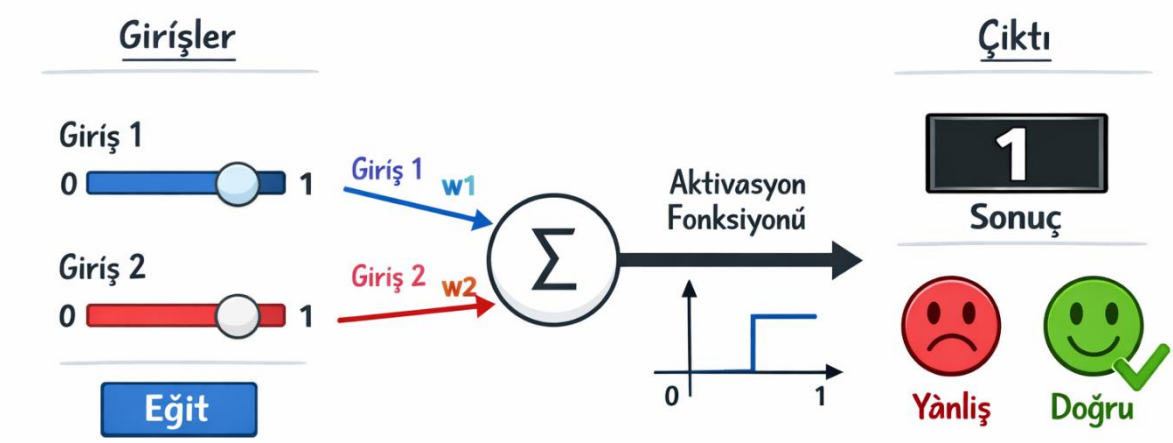
Perceptronun en dikkat çekici özelliği, *mantık kapılarını öğrenebilmesidir***.** Yani bilgisayar, AND, OR ve NOT gibi temel mantık işlemlerini perceptronlar aracılığıyla uygulayabilir. Bu, yapay zekânın öğrenme sürecinin ilk basamağıdır. Perceptronun amacı, giriş verilerinden bir örüntü tanımak ve doğru kararı vermektir.

Diyelim ki bir makineye el yazısı rakamları tanıma görevi veriliyor. Girdi olarak bir 3x3 piksel kareyi kullanıyoruz (toplam 9 piksel). Her piksel ya siyah (1) ya beyaz (0) olabilir. Yapay sinir hücresi, her piksele ağırlık atar ve hepsini toplar. Eğer toplam belirli bir değeri aşarsa bu hücre, örneğin "1" rakamını temsil ettiğini tahmin eder.

Bu örnek, perceptronun mantığını anlamak için oldukça basittir ama karmaşık verilerle katman katman birleştirildiğinde inanılmaz bir öğrenme kapasitesine ulaşır.

Bir başka örnek günlük hayattan: Bir e-posta filtreleme sistemi düşünün. Gelen e-postadaki kelimeler (girişler) perceptron tarafından değerlendirilir. Eğer spam olma olasılığı belirli bir eşik değerini aşarsa e-posta spam kutusuna yönlendirilir.

Şekil 7. Perceptron



Biyolojik beyin ve yapay nöron arasındaki benzerlik burada yatmaktadır: her iki sistem de *girdi → işlem → çıktı* mantığı ile çalışır. Farklı olan, yapay sinir ağlarının çok daha hızlı ve büyük veri setlerinde çalışabilmesidir.

Perceptron, yapay sinir ağının temel yapı taşıdır. Python ile basit bir perceptron modeli kurabiliriz. Perceptronun nasıl çalıştığını daha iyi anlamak için basit bir örnek üzerinden ilerleyelim: *İki girişli AND kapısı.*

Giriş X1	**Giriş X2**	**Çıkış Y (AND)**
0	0	0
0	1	0
1	0	0
1	1	1

Bu tablo, AND mantığını gösterir: yalnızca her iki giriş de 1 olduğunda çıkış 1 olur. Perceptron, bu mantığı öğrenmek için girişlerin ağırlıklarını ve eşik değerini ayarlayabilir. Öğrenme süreci şu şekilde işler:

1. Perceptron, başlangıçta rastgele ağırlıklarla başlar.
2. Her eğitim örneği için, perceptron çıktıyı hesaplar ve gerçek çıktı ile karşılaştırır.
3. Hata varsa, perceptron ağırlıkları ve eşik değerini düzeltir.
4. Bu süreç, tüm tablo doğru sonuçları üreten ağırlıklar bulunana kadar devam eder.

Örneğin, başlangıçta X1 ve X2 girişlerinin ağırlıkları 0.5 ve eşik değeri 0.8 olabilir. İlk örnekte (X1=1, X2=1), ağırlıklı toplam 1.0 olur ve eşik değerini geçtiği için doğru şekilde 1 çıktısı alınır. Başka örneklerde hata oluşursa, ağırlıklar güncellenir ve perceptron doğru öğrenmeye başlar.

Bu örnek, perceptronun temel öğrenme mantığını ve bilgisayarın nasıl karar aldığını gösterir. Basit mantık kapılarından başlayarak perceptron, daha karmaşık görevleri öğrenmek için bir temel oluşturur. Gerçek hayatta perceptronlar, görüntü tanıma, ses tanıma ve dil işleme gibi alanlarda daha karmaşık ağların parçaları hâline gelir.

Perceptron, yapay sinir ağlarının temel taşıdır. Basit yapısı sayesinde, bilgisayarlara *öğrenme yeteneği* kazandırır. Günümüzde perceptronlar, tek başına büyük uygulamalarda kullanılmasa da, çok katmanlı yapay sinir ağlarının yapı taşını oluşturur.

Uygulama örnekleri:

- **Mantık kapıları ve temel karar sistemleri:** Bilgisayarın karar alma sürecini öğrenmek için kullanılır.
- **Görüntü tanıma:** Örneğin bir perceptron, bir resimdeki siyah ve beyaz pikselleri değerlendirerek basit desenleri tanıyabilir.
- **Ses ve konuşma tanıma:** Perceptron, temel ses özelliklerini analiz ederek sesli komutları tanıyabilir.

Perceptronlar, insan beyninin nasıl çalıştığını anlamak ve bunu bilgisayar ortamına taşımak için geliştirilmiş basit ama güçlü araçlardır. Öğrenme sürecini adım adım gösterir ve derin öğrenmenin temelini atar.

Katmanlar ve Ağ Yapısı

Yapay sinir ağları, bilgisayarların verileri işleyip öğrenmesini sağlayan en temel yapılar arasında yer alır. Bu ağlar, tıpkı insan beynindeki nöronlar gibi birbirine bağlı bilgi işleme birimlerinden oluşur. Yapay sinir ağlarının çalışma mantığını anlamak için öncelikle *katman kavramını* bilmek gerekir. Bir yapay sinir ağı, üç ana bölümden oluşur*:* giriş katmanı, gizli katmanlar ve çıkış katmanı. Bu bölümde her katmanın işlevini, nasıl çalıştığını ve birbirleriyle ilişkilerini ayrıntılı şekilde inceleyeceğiz.

Giriş Katmanı

Giriş katmanı, verinin ağın içine girdiği ilk noktadır. Bu katman, bilgisayarın dış dünyadan aldığı bilgiyi alır ve işleme sürecine başlar. Giriş katmanındaki her bir nöron, ağın göreceği bir veriye karşılık gelir.

Örneğin, bir *el yazısı rakam tanıma* uygulamasını düşünelim. Bu uygulamada bilgisayarın tanıyacağı her rakam, 28x28 piksel boyutunda bir resim olarak temsil edilir. Her bir piksel, gri tonlarında bir değer taşır; örneğin siyah bir piksel 0, beyaz bir piksel 255 değerinde olabilir. Bu 28x28 pikselin her biri, giriş katmanındaki bir nörona bağlanır. Böylece ağ, rakamı oluşturan tüm pikselleri ayrı ayrı görebilir ve işleyebilir.

Giriş katmanı, temel olarak *veriyi ağın anlayabileceği bir formata dönüştürme* işlevi görür. Bu aşamada veriye herhangi bir karar verme veya tahmin yeteneği eklenmez; katman sadece ham

veriyi alır ve sonraki katmanlara iletir. Yani giriş katmanı, bir sınıfın öğrencilerini ders öğretmenine sırayla tanıtan bir görevlidir; öğretmen her öğrenciyi tek tek inceler ve sonraki değerlendirme katmanına aktarır.

Gizli Katmanlar

Giriş katmanından sonra veriler *gizli katmanlar* aracılığıyla işlenir. Gizli katmanlar, yapay sinir ağlarının en önemli kısmıdır çünkü burada veri içerisindeki karmaşık desenler ve örüntüler öğrenilir. Gizli katmanlar olmadan ağ, yalnızca basit kuralları uygulayabilir; karmaşık ilişkileri öğrenemez.

Bir gizli katmandaki nöron sayısı arttıkça, ağın öğrenme kapasitesi ve karmaşık ilişkileri tanıma yeteneği de artar. Her bir nöron, kendisine bağlı girişlerden aldığı ağırlıklı değerleri toplar, belirli bir eşik değere göre işler ve sonucu bir sonraki katmana iletir. Bu işlem, bir zincirleme mantıkla çalışır: bir katmandaki nöronlar basit özellikleri algılarken, sonraki katmanlar bu basit özellikleri birleştirerek daha karmaşık ve anlamlı desenler çıkarır.

El yazısı rakam tanıma örneğinde bunu şöyle düşünebiliriz:

- **Birinci gizli katman:** Rakamın kenarlarını, çizgilerini ve basit şekillerini tanır. Örneğin, bir "7" rakamının üst çizgisini veya alt çizgisini ayırt edebilir.
- **İkinci gizli katman:** Birinci katmandan gelen bilgileri birleştirerek rakamın tamamını tanımaya başlar. Örneğin, çizgilerin birleşimi "7" rakamını oluşturur ve diğer rakamlardan ayırt edilebilir hâle gelir.
- **Üçüncü gizli katman (varsa):** Daha karmaşık özellikleri öğrenir ve rakamın farklı yazım şekillerini tanıyabilir. Örneğin, bir kişinin "7" yazarken hafifçe eğik yazması bile bu katmanda doğru şekilde işlenir.

Gizli katmanlar, bir yapay sinir ağını *basit perceptronlardan çok daha güçlü* hâle getirir. Her katman, veriyi bir seviyeden diğerine taşıyarak öğrenmeyi kademeli bir şekilde derinleştirir. Bu yüzden çok katmanlı ağlara *derin sinir ağı (Deep Neural Network)* denir. İnsan beyninde de düşük seviyeli nöronlar kenar ve renkleri tanırken, yüksek seviyeli nöronlar nesneleri ve anlamları tanır; yapay sinir ağlarındaki gizli katmanlar tam olarak bu mantığı taklit eder.

Çıkış Katmanı

Gizli katmanlarda veri işlendiğinde, son adım olarak çıkış katmanı devreye girer. Çıkış katmanı, ağın öğrendiği bilgiyi somut bir sonuca dönüştürür. Bu katmandaki her bir nöron, ağın tahmin edebileceği farklı sonuçları temsil eder.

El yazısı rakam tanıma örneğini ele alırsak, çıkış katmanı 10 nörondan oluşur ve her nöron sırasıyla 0'dan 9'a kadar olan rakamları temsil eder. Ağın çıkışı, hangi rakamın en yüksek olasılıkla doğru olduğuna göre belirlenir. Örneğin, bir görüntüdeki rakamın "3" olduğuna karar verildiğinde, çıkış katmanındaki "3" nöronu en yüksek değeri alır ve tahmin olarak bu rakam seçilir.

Çıkış katmanı, tüm gizli katmanlardan gelen bilgiyi toplar, değerlendirir ve nihai karar veya tahmini verir. Buradaki süreç, bir sınav sonucunu hesaplayan öğretmen gibi düşünülebilir: tüm ödevleri ve sınavları değerlendirir, notları toplar ve öğrenciye final notunu verir. Yani çıkış katmanı, ağın öğrenme sürecinin "son karar" noktasıdır.

Giriş katmanı, gizli katmanlar ve çıkış katmanı bir zincirleme mantıkla birbirine bağlıdır. Veri giriş katmanına gelir, gizli katmanlarda işlenir ve çıkış katmanı ile sonuca ulaşır. Bu süreç, *ileri beslemeli ağ (feedforward network)* olarak adlandırılır; çünkü veri bir yönde akar ve sürekli ileri taşınır.

Ağ, bu üç katmanın birlikte çalışması sayesinde öğrenebilir, tahmin yapabilir ve yeni verilerle karşılaştığında doğru kararlar üretebilir. Giriş katmanı veriyi alır, gizli katmanlar örüntüleri çıkarır ve çıkış katmanı sonuçları üretir. Bu yapı, basit bir karar sistemi için yeterli olabilir, ancak daha karmaşık görevler için birden fazla gizli katman ve daha fazla nöron gerekir.

Basit Perceptron ile Mantık Kapıları

Perceptron, yapay sinir ağlarının en temel birimidir. Önceki bölümlerde öğrendiğimiz gibi, perceptron bir nöron gibi çalışır: girişleri alır, ağırlıklarla çarpar, toplar ve eşik değerine göre bir çıktı üretir. Bu basit yapı sayesinde perceptron, *AND, OR ve NOT gibi mantık kapılarını* öğrenebilir.

Perceptron, yapay sinir ağlarının en temel yapı taşıdır ve aslında bilgisayarların öğrenmesini sağlayan ilk basit nöron modeli olarak düşünülebilir. İnsan beynindeki bir nöronun işleyişinden esinlenilmiştir. Beynimizde nöronlar, dendritlerden aldıkları sinyalleri toplar ve belirli bir eşik değerini aştığında akson aracılığıyla diğer nöronlara sinyal gönderir. Perceptron da aynı mantığı bilgisayar ortamına taşır: girişlerden gelen değerleri alır, her birini bir ağırlıkla çarpar, sonuçları toplar ve bu toplam belirli bir eşik değerini geçerse bir çıktı üretir. Eğer eşik değeri aşılmazsa çıktı üretilmez veya farklı bir değer alınır. Bu mekanizma, perceptronun "öğrenmesini" ve karar vermesini sağlayan temel prensiptir.

Perceptronun en temel kullanım alanlarından biri *mantık kapılarıdır.* Mantık kapıları, bilgisayarlarda ve elektronik devrelerde kullanılan temel birimlerdir ve basit "evet/hayır" ya da "1/0" kararlarını temsil eder. Örneğin, AND kapısı, yalnızca iki girişin de 1 olduğu durumda 1 çıktısı verir; OR kapısı, girişlerden en az biri 1 olduğunda 1 çıktısı verir; NOT kapısı ise girişi tersine çevirir. Perceptron, ağırlıklarını ve eşik değerini doğru şekilde ayarlayarak bu kapıların davranışını öğrenebilir. Bu sayede, bilgisayar basit karar mekanizmalarını kavrayabilir ve temel mantık işlemlerini uygulayabilir.

Perceptronun önemli bir avantajı, *basit olmasına rağmen öğrenme yeteneğine sahip olmasıdır.* Girdi değerleri ve beklenen çıktılar verildiğinde, perceptron ağırlıklarını adım adım değiştirerek doğru sonuçlara ulaşmayı öğrenir. Örneğin, bir AND kapısı için ağırlıkları ve eşik değerini başlangıçta rastgele belirleyip, hatalı çıktı gördükçe ağırlıkları güncelleyebilir. Zamanla, perceptron tüm giriş kombinasyonları için doğru çıktıyı vermeyi öğrenir.

Bu özellik, perceptronu hem eğitim amaçlı bir araç hem de daha karmaşık yapay sinir ağlarının temel yapı taşı hâline getirir. Basit bir perceptron ile başlamak, öğrencilerin ve araştırmacıların yapay sinir ağlarının mantığını kavramasına yardımcı olur. Perceptron sayesinde bilgisayarlar, temel mantık kapılarını öğrenebilir ve bu küçük adımlar, daha karmaşık karar süreçlerini gerçekleştirebilen derin öğrenme ağlarının temelini oluşturur.

Örnek: AND Kapısı

Bir AND kapısı, iki girişin de 1 olduğu durumda 1 çıktısı verir; diğer durumlarda çıktı 0'dır. Tablo şöyle görünür:

Giriş X1	**Giriş X2**	**Çıkış Y (AND)**
0	0	0
0	1	0
1	0	0
1	1	1

Adım Adım Uygulama:

1. **Girişleri Belirle:** X1 ve X2 giriş değerlerini 0 veya 1 olarak al.
2. **Çıkışları Belirle:** AND kapısı için yalnızca 1+1 girişinde çıktı 1 olur; diğer durumlarda çıktı 0'dır.
3. **Ağırlıkları ve Eşik Değerini Ayarla:** Her girişe bir ağırlık ata ve bir eşik değeri belirle. Örneğin, X1 ve X2 için ağırlık 0,6 ve eşik 1,0 olabilir.
4. **Hataları Düzelt ve Tekrar Deneyin:** Perceptron, ilk denemede hatalı sonuç verebilir. Bu durumda ağırlıkları ve eşik değerini güncelle. Bu süreç, perceptronun **öğrenmesini** sağlar.

Bu küçük uygulama, perceptronun temel mantığını anlamak için idealdir. Öğrenciler, giriş ve çıkışları değiştirerek farklı mantık kapılarını (OR, NOT) da deneyebilir ve perceptronun nasıl çalıştığını deneyimleyebilir.

Basit perceptron uygulamaları, öğrencilere yapay sinir ağlarının *temel öğrenme mekanizmasını* gösterir. Bu deney, ileri düzey derin öğrenme modellerini anlamanın temelini oluşturur. Basit bir mantık kapısını öğrenebilen bir perceptron, daha karmaşık örüntüleri tanıyabilen büyük ağların yapı taşını oluşturur.

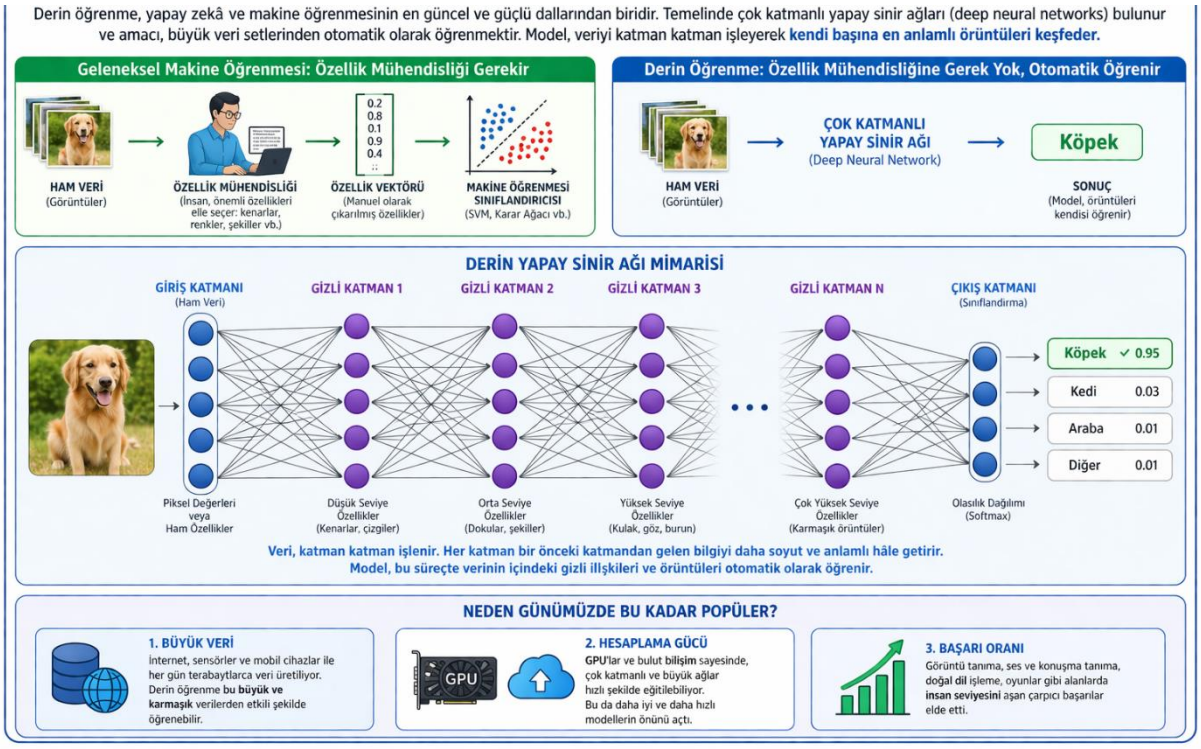
Derin Öğrenme Nedir?

Yapay sinir ağlarının temelinde *perceptronlar* ve tek katmanlı ağlar yatar. Ancak gerçek dünyadaki veriler genellikle çok karmaşıktır. Bir fotoğraftaki yüzü tanımak, bir konuşmayı yazıya

dönüştürmek veya bir el yazısı rakamı doğru şekilde sınıflandırmak, tek bir katmanla yapılamayacak kadar zordur. İşte bu noktada *derin öğrenme* devreye girer.

Derin öğrenme, birden fazla *gizli katmana* sahip yapay sinir ağlarını tanımlamak için kullanılan genel bir terimdir. Bu ağlar, veriyi katman katman işler ve her katman, verinin farklı bir özelliğini öğrenir. Katman sayısı arttıkça, ağ daha karmaşık ve detaylı ilişkileri ve örüntüleri öğrenebilir. Bu nedenle derin öğrenme, bilgisayarların insan benzeri şekilde düşünmesini ve karar vermesini sağlayan güçlü bir yöntemdir.

Şekil 5. Derin Öğrenme Nedir?



Derin öğrenme, son yıllarda özellikle popüler hâle gelmiştir ve bunun birkaç nedeni vardır:

1. **Büyük veri ile çalışabilir:** Günümüzde her saniye internete milyarlarca fotoğraf, video ve ses kaydı yükleniyor. Derin öğrenme ağları, bu büyük veri kümeleri ile çalışabilir ve içlerinden anlamlı örüntüler çıkarabilir. Örneğin, sosyal medyada paylaşılan milyonlarca fotoğraf üzerinden yüz tanıma algoritmaları geliştirilebilir.

2. **Karmaşık desenleri ve örüntüleri öğrenebilir:** Basit bir perceptron veya tek katmanlı ağ, yalnızca temel mantık kurallarını ve küçük örüntüleri öğrenebilir. Ancak derin öğrenme ağları, birden fazla gizli katman sayesinde karmaşık örüntüleri tanıyabilir. Örneğin, bir fotoğraftaki bir kişinin yüzündeki göz, burun ve ağız gibi farklı özellikleri birbirine bağlayarak, kişiyi doğru şekilde tanıyabilir.

3. **İnsan müdahalesi olmadan özellik çıkarabilir:** Geleneksel makine öğrenmesi yöntemlerinde, veri önceden işlenmeli ve önemli özellikler manuel olarak seçilmelidir. Ancak derin öğrenme ağları, verinin kendisinden önemli özellikleri kendileri çıkarabilir.

Örneğin, bir el yazısı rakam tanıma sisteminde, ağ hangi çizgilerin ve kenarların rakamı tanımlamak için önemli olduğunu otomatik olarak öğrenir. Bu, insanın müdahalesi olmadan öğrenmeyi mümkün kılar ve büyük veri ile çalışmayı çok daha pratik hâle getirir.

4. **Çeşitli uygulama alanlarına uygundur:** Derin öğrenme, görüntü tanıma, ses tanıma, doğal dil işleme, tıbbi görüntü analizi, sürücüsüz araçlar ve oyun oynama gibi çok farklı alanlarda başarıyla kullanılmaktadır. Bu geniş uygulama yelpazesi, derin öğrenmenin neden bu kadar önem kazandığını açıklar.

Özetle, derin öğrenme, bilgisayarların karmaşık verileri analiz etmesine, anlam çıkarmasına ve doğru kararlar vermesine olanak tanır. Günümüzde yapay zekâ uygulamalarının çoğu, derin öğrenme tekniklerini temel alır.

Derin öğrenme ağlarının katmanlı yapısı, insan beyninin işleyişine oldukça benzer. Beynimiz, çevremizden gelen bilgiyi katman katman işler. Örneğin, bir görsel bilgi geldiğinde beynimiz önce *kenarları* algılar, sonra *basit şekilleri* tanır ve en sonunda bu şekilleri birleştirerek *nesneleri ve anlamları* çıkarır. Bu katmanlı işleyiş, derin öğrenme ağlarının mantığıyla neredeyse aynıdır.

Düşük seviyeli katmanlar, verideki *temel özellikleri* öğrenir: bir resimdeki dikey ve yatay çizgiler, renk tonları veya basit desenler. Orta seviyeli katmanlar, bu temel özellikleri birleştirerek daha *karmaşık yapılar* oluşturur: bir harf şekli, bir göz, bir burun gibi. Üst seviyeli katmanlar ise, bu karmaşık yapıları birleştirerek *tüm nesneyi veya anlamı* tanır: bir rakam, bir yüz veya bir araba gibi.

Bu yapı, insan beyninin bilgi işleme süreciyle paralellik gösterir ve derin öğrenme ağlarını özellikle görsel ve işitsel verilerde etkili kılar. İnsan beyninin katman katman öğrenmesi, bilgisayarlara ilham vermiştir ve bu sayede bilgisayarlar, önceki yıllarda mümkün olmayan görevleri başarabilir hâle gelmiştir.

Derin Öğrenmenin Öğrenme Süreci

Derin öğrenme ağları, veriyi ileri doğru aktarır *(feedforward)* ve ardından hatayı geri yayarak *(backpropagation)* ağırlıkları günceller. Bu süreç, ağın doğru karar vermesini sağlayan temel mekanizmadır.

1. **İleri Besleme (Feedforward):**
   Girdi verileri giriş katmanına gelir ve her katmandaki nöronlar aracılığıyla işlenir. Her gizli katman, veriyi bir sonraki katmana taşır. Çıkış katmanında ağın tahmini elde edilir.

2. **Hata Hesaplama:**
   Ağın tahmini, gerçek değerle karşılaştırılır ve hata hesaplanır. Örneğin, bir el yazısı rakam tanıma modelinde, görüntüdeki rakam “3” iken ağ “5” tahmini yaptıysa hata oluşur.

3. **Geri Yayılım (Backpropagation):**
   Hata, ağ boyunca geriye doğru yayılır ve her nöronun ağırlıkları bu hatayı azaltacak şekilde güncellenir. Bu işlem, tıpkı insan beyninin deneyimlerden öğrenmesi gibi ağın kendini düzeltmesini sağlar.

4. **Tekrar:**
   Bu işlem, ağ doğru tahminler yapana kadar veya belirli bir sayıdaki eğitim örneği işlenene kadar tekrar edilir. Zamanla ağ, verinin örüntülerini öğrenir ve yeni verilerle karşılaştığında doğru tahminler yapabilir.

Derin Öğrenmenin Uygulama Örnekleri

Derin öğrenme, günlük yaşamda birçok alanda kullanılmaktadır. Bazı örnekler şunlardır:

- **Görüntü tanıma:** Facebook veya Google Fotoğraflar, milyonlarca görüntüden yüzleri tanıyabilir.
- **Ses tanıma:** Akıllı asistanlar (Siri, Alexa) konuşulan kelimeleri anlayabilir.
- **Tıbbi görüntü analizi:** MRI ve röntgen görüntüleri analiz edilerek hastalıklar erken teşhis edilebilir.
- **Otomatik çeviri:** Dil modelleri, farklı dillerdeki cümleleri anlamlı şekilde çevirebilir.
- **Sürücüsüz araçlar:** Kameralardan gelen görüntüleri analiz ederek çevredeki nesneleri tanır ve güvenli sürüş sağlar.

Bu örnekler, derin öğrenmenin neden çağımızın en önemli teknolojilerinden biri olduğunu gösterir.

Derin Öğrenme ile El Yazısı Rakam Tanıma

Perceptron ile mantık kapıları örneği, yapay sinir ağlarının temelini gösterirken, derin öğrenme örnekleri bu kavramları gerçek dünyadaki problemlere uygulamanın yolunu gösterir. El yazısı rakam tanıma, derin öğrenmenin klasik ve anlaşılır bir örneğidir.

Veri Seti: MNIST

MNIST, el yazısı rakamları içeren bir veri setidir. Her rakam 28x28 piksel boyutunda siyah-beyaz bir görüntü olarak temsil edilir. Bu veri seti, bilgisayarlara rakamları tanımayı öğretmek için kullanılır ve dünya çapında derin öğrenme eğitimlerinde standart bir örnek olarak kabul edilir.

Ağ Yapısı

1. **Giriş Katmanı:** 28x28 piksel, yani 784 giriş nöronu. Her bir piksel, bir nörona karşılık gelir.
2. **Gizli Katmanlar:** 128 veya daha fazla nöron içeren bir veya daha fazla gizli katman. Bu katmanlar, rakamın çizgilerini, kenarlarını ve şekillerini öğrenir.

3. **Çıkış Katmanı:** 10 nöron, her biri 0–9 rakamlarından birini temsil eder. Ağın çıktısı, hangi rakamın doğru olduğuna karar verir.

Öğrenme Süreci

Ağ, **ileri besleme (feedforward)** ve **geri yayılım (backpropagation)** yöntemlerini kullanarak öğrenir:

1. **İleri Besleme:** Piksel değerleri giriş katmanına gelir ve gizli katmanlardan geçerek çıkış katmanına ulaşır.
2. **Hata Hesaplama:** Ağın tahmini, gerçek rakam ile karşılaştırılır. Örneğin, görüntüdeki rakam 3 iken ağ 5 tahmin ettiyse, hata oluşur.
3. **Geri Yayılım:** Hata, ağ boyunca geriye doğru yayılır ve nöron ağırlıkları, hatayı azaltacak şekilde güncellenir.
4. **Tekrar:** Bu süreç, ağ doğru tahminler yapana kadar tekrar edilir. Böylece ağ, rakamların temel örüntülerini öğrenir.

Örnek Çıktı

Bir el yazısı rakam görüntüsü ağa verildiğinde, ağ şöyle çalışır:

- Giriş katmanı pikselleri alır.
- İlk gizli katman kenarları ve çizgileri tanır.
- İkinci gizli katman, bu çizgileri birleştirerek rakamın tamamını tanır.
- Çıkış katmanı hangi rakam olduğunu belirler ve sonucu verir: “Bu rakam 7.”

Bu süreç, perceptron mantığını genişleterek daha karmaşık veriler üzerinde uygulanmasını gösterir. Basit mantık kapılarından başlayıp, yüzlerce nöron ve katman içeren derin ağlara geçiş yapılabilir.

Mini Uygulamanın Faydaları

1. **Teoriyi Pratiğe Dönüştürme:** Öğrenciler, perceptron ve derin öğrenme kavramlarını kendi deneyleri ile pekiştirir.
2. **Görselleştirme:** Mantık kapıları veya el yazısı rakam tanıma örnekleri, ağın nasıl çalıştığını somut biçimde gösterir.
3. **Küçük Adımlarla Öğrenme:** Basit perceptron örnekleri, karmaşık derin öğrenme modellerine geçişi kolaylaştırır.
4. **Hatalardan Öğrenme:** Ağın hataları düzeltmesi, öğrencilerin öğrenmenin temel prensibini anlamasını sağlar.

Bölüm Özeti

Bu bölümde, yapay zekânın en güçlü tekniklerinden biri olan yapay sinir ağları ve onun gelişmiş hâli olan derin öğrenme ele alınmıştır. Yapay sinir ağları, insan beynindeki nöronlardan esinlenerek geliştirilmiş matematiksel modellerdir ve verilerden öğrenerek tahmin yapma yeteneğine sahiptir.

Bölüm, en temel yapı taşı olan perceptron ile başlar. Perceptron, bir yapay sinir hücresidir; giriş verilerini alır, ağırlıklarla çarpar, toplar ve bir eşik değerine göre çıktı üretir. Bu basit yapı, AND gibi mantık kapılarını modelleyebilir ve yapay sinir ağlarının temelini oluşturur.

Daha karmaşık problemleri çözebilmek için perceptronlar bir araya getirilerek katmanlı yapılar oluşturulur. Bu yapılar üç ana bölümden oluşur: giriş katmanı (verinin modele girdiği yer), gizli katmanlar (bilginin işlendiği ara katmanlar) ve çıkış katmanı (sonucun üretildiği katman). Gizli katmanların sayısı arttıkça modelin öğrenme kapasitesi de artar.

Bu noktada derin öğrenme kavramı devreye girer. Derin öğrenme, çok sayıda gizli katmana sahip sinir ağlarını ifade eder ve özellikle karmaşık veri türlerinde (görüntü, ses, metin) oldukça başarılıdır. Derin öğrenme sürecinde model, veriler üzerinden tahminler yapar, hatalarını hesaplar ve bu hataları azaltmak için kendini sürekli günceller. Bu süreç tekrarlandıkça model daha doğru sonuçlar üretir.

Bölümde ayrıca derin öğrenmenin gerçek hayattaki uygulamalarına da değinilmiştir. Özellikle el yazısı rakam tanıma örneği, modelin görüntüleri analiz ederek sayıları doğru şekilde sınıflandırabildiğini gösterir. Bu tür uygulamalar, bankacılıktan posta sistemlerine kadar birçok alanda kullanılmaktadır.

Sonuç olarak bu bölüm, yapay sinir ağlarının temel mantığını, katmanlı yapılarını ve derin öğrenmenin nasıl çalıştığını sezgisel bir şekilde açıklayarak, modern yapay zekâ sistemlerinin arkasındaki temel prensipleri ortaya koymaktadır.

Soru-Cevap

1. Yapay sinir ağı nedir?
Yapay sinir ağı, insan beynindeki nöronlardan esinlenerek oluşturulan ve verilerden öğrenebilen matematiksel bir modeldir.

2. Perceptron nedir?
Perceptron, yapay sinir ağlarının en basit birimi olan yapay nörondur; girişleri alır, işler ve bir çıktı üretir.

3. Perceptron nasıl çalışır?
Giriş verilerini ağırlıklarla çarpar, toplar ve bir eşik değerine göre sonucu 0 veya 1 olarak belirler.

4. Yapay sinir ağlarında katmanlar nelerdir?
Giriş katmanı, gizli katmanlar ve çıkış katmanı olmak üzere üç temel katman bulunur.

5. Giriş katmanının görevi nedir?
Verilerin modele ilk girdiği katmandır ve ham bilgiyi sisteme aktarır.

6. Gizli katmanlar ne işe yarar?
Veriyi işler, örüntüleri öğrenir ve daha karmaşık ilişkileri ortaya çıkarır.

7. Çıkış katmanı ne yapar?
Modelin nihai tahminini veya sonucunu üretir.

8. Derin öğrenme nedir?
Birden fazla gizli katmana sahip yapay sinir ağlarının kullanıldığı gelişmiş bir makine öğrenmesi yöntemidir.

9. Derin öğrenme nasıl öğrenir?
Tahmin yapar, hatayı hesaplar ve bu hatayı azaltacak şekilde kendini sürekli günceller.

10. Derin öğrenmenin günlük hayattaki bir uygulaması nedir?
El yazısı rakam tanıma sistemleri, derin öğrenme kullanarak sayıları doğru şekilde tanıyabilir.

ÜRETKEN YAPAY ZEKÂ VE BÜYÜK DİL MODELLERİ

Yapay zekânın son yıllardaki en heyecan verici alanlarından biri, üretken yapay zekâdır. Bu sistemler, insanın hayal gücüne yakın biçimde yeni içerikler üretebilir; metin, görsel, müzik ve hatta video yaratabilirler. Üretken yapay zekâ, yalnızca bir hesap makinesi gibi işlem yapmakla kalmaz; adeta bir yazar, ressam veya besteci gibi düşünebilir. Ancak bu "düşünme" insanınkinden farklıdır; istatistiksel örüntüler üzerinden çalışır, geçmiş deneyimlerinden öğrenir ve olasılık hesapları ile yeni içerik üretir. Bu bölümde, büyük dil modellerinin mantığını, çalışma prensiplerini, metin ve görsel üretim örneklerini, halüsinasyon problemini ve prompt yazmanın inceliklerini ele alacağız.

Üretken Yapay Zekâ (Generative AI)

Üretken yapay zekâ (Generative AI), yapay zekânın en heyecan verici ve yaratıcı alanlarından biridir. Bu teknoloji, var olan verilerden öğrenerek tamamen yeni içerikler üretebilen sistemleri ifade eder. Buradaki "üretken" kelimesi, sistemin yalnızca analiz yapmak veya sınıflandırmakla kalmayıp, özgün ve önce görülmemiş içerikler yaratabilme yeteneğini vurgular. Üretken yapay zekâ, metin, görüntü, ses ve hatta üç boyutlu modeller oluşturabilir. Günümüzde en popüler örnekler arasında DALL-E, Midjourney ve büyük dil modelleri (LLM – GPT-4, LLaMA, Claude) yer alır. Bu modeller, hem sanat ve tasarım dünyasında hem de eğitim, bilim, tıp ve mühendislik gibi alanlarda devrim niteliğinde uygulamalara imkân tanımaktadır.

Görüntü üretimi açısından DALL-E ve Midjourney öne çıkar. DALL-E, OpenAI tarafından geliştirilen bir sistem olup, kullanıcıdan aldığı metin açıklamasına dayanarak tamamen özgün bir resim üretir. Örneğin "uzayda kahve içen bir robot" dediğinizde, model bu metni analiz eder, kavramları çözer ve bunları birleştirerek özgün bir görsel oluşturur. Midjourney ise özellikle sanatsal ve estetik açıdan yüksek çözünürlüklü görüntüler üretmekte öne çıkar; sanatçılar ve tasarımcılar için yaratıcı bir araç hâline gelmiştir. Bu tür üretken sistemler, kullanıcıya sadece hazır içerik sunmakla kalmaz, aynı zamanda hayal gücünü genişleten, ilham veren bir rol oynar. Örneğin bir moda tasarımcısı, Midjourney'e "geleceğin sokak modası, neon renkler ve holografik kumaşlar" komutunu verdiğinde, sistem ortaya tamamen yeni ve yaratıcı tasarım fikirleri çıkarabilir.

Metin üretimi açısından ise GPT-4, ChatGPT ve benzeri büyük dil modelleri öne çıkar. Bu modeller, devasa metin verileri üzerinde eğitilmiştir ve dilin yapısını, bağlamını ve mantığını öğrenmiştir. Örneğin bir öğrenci "Ortaçağda geçen bir bilim kurgu hikayesi yaz" dediğinde, model tarihsel bağlam, karakter özellikleri ve kurgusal öğeleri bir araya getirerek tamamen yeni bir hikâye üretir. Üretken yapay zekâ, sadece hikâye üretmekle kalmaz; aynı zamanda metin özetleme, soru yanıtlama, çeviri ve hatta şiir veya şarkı sözü yazma gibi yaratıcı görevlerde de kullanılabilir. Örneğin bir eğitim uygulamasında, öğrenci kısa bir metni girer ve model bunu daha anlaşılır bir özet hâline getirir veya belirli bir tonla yeniden yazar.

Python ile mini bir metin üretimi örneği, bu sürecin mantığını anlamak için kullanılabilir:

---

1 Kütüphaneleri yükleme

```
from transformers import GPT2LMHeadModel, GPT2Tokenizer
```

- **GPT2Tokenizer**: Metni modelin anlayacağı **token'lara (sayısal kodlara)** çevirir ve üretilen token'ları tekrar metne dönüştürür.
- **GPT2LMHeadModel**: GPT-2 modelini yükler; **LMHead** (Language Modeling Head) eklenmiş, yani metin üretmeye hazır.

2 Model ve tokenizer'ı yükleme

```
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
model = GPT2LMHeadModel.from_pretrained("gpt2")
```

- from_pretrained("gpt2") → önceden eğitilmiş **GPT-2 küçük modelini** indirir ve hazır hale getirir.
- tokenizer → metni token'lara çevirir.
- model → token'lar üzerinden tahminler yaparak yeni metin üretebilir.

3 Başlangıç metni oluşturma

```
input_text = "Uzayda bir macera başlıyor: "
inputs = tokenizer.encode(input_text, return_tensors='pt')
```

- input_text: üretimin başlayacağı metin (prompt).
- tokenizer.encode → metni **token ID'lerine çevirir**.
- return_tensors='pt' → PyTorch tensörüne çevirir, model PyTorch formatında çalışır.

Örnek: "Uzayda bir" → [1234, 5678, ...] gibi bir sayı dizisi.

4 Metin üretimi

outputs = model.generate(inputs, max_length=100, do_sample=True)

- model.generate → GPT-2 ile yeni metin üretir.
- **inputs** → başlangıç token'ları.
- **max_length=100** → üretilen metnin toplam token sayısı 100'ü geçmez.
- **do_sample=True** → rastgele örnekleme ile farklı metinler üretilmesini sağlar (aksine greedy seçerse hep aynı metin çıkar).

5️⃣ Üretilen metni yazdırma

print(tokenizer.decode(outputs[0]))

- outputs[0] → üretilen token ID dizisi.
- tokenizer.decode → token'ları tekrar **insan okunabilir metne** çevirir.
- Sonuç: "Uzayda bir macera başlıyor: ..." ile devam eden yeni bir metin.

💡 Özet Mantık:

1. Başlangıç metni **token'lara** çevrilir.
2. GPT-2 modeli token'ları kullanarak **sonraki token'ları tahmin eder**.
3. Tahmin edilen token'lar tekrar metne çevrilir ve üretilen metin elde edilir.

Bu örnek, GPT-2 modelinin verilen metni alıp devam ettirerek özgün bir içerik üretmesini gösterir. Öğrenciler ve araştırmacılar, farklı başlangıç metinleri deneyerek modelin nasıl yeni cümleler ve fikirler oluşturduğunu gözlemleyebilir.

Üretken yapay zekânın uygulama alanları yalnızca eğlence veya sanatla sınırlı değildir. Tasarım, moda, eğitim, tıp ve bilimsel araştırmalarda da devrim niteliğinde uygulamalara imkân verir. Örneğin tıp alanında, üretken modeller ilaç moleküllerini tasarlamak veya protein yapılarını tahmin etmek için kullanılabilir. Bir araştırmacı, belirli bir hastalığa yönelik potansiyel molekülleri model üzerinde deneyerek ilaç geliştirme sürecini hızlandırabilir. Benzer şekilde biyoinformatik alanında, protein katlanma ve etkileşim modelleri GPT tabanlı sistemler veya benzeri üretken modellerle tahmin edilebilir.

Üretken yapay zekâ ayrıca ses ve müzik üretiminde de kullanılır. Örneğin bir müzik uygulaması, kullanıcıdan birkaç nota veya tarz bilgisi alarak yeni bir melodiyi otomatik olarak üretebilir. Sesli asistanlar ve oyun endüstrisinde de karakterlerin konuşma ve ses efektlerini üretmek için bu teknoloji kullanılmaktadır. Örneğin bir oyun geliştiricisi, karakterin farklı durumlarda söyleyeceği diyalogları üretmek için GPT veya benzeri bir üretken model kullanabilir.

Sanat ve pazarlama alanında üretken yapay zekâ, içerik üretiminde maliyetleri azaltır ve hız kazandırır. Reklam ajansları, yeni kampanya görselleri veya sosyal medya içerikleri üretmek için Midjourney veya DALL-E'yi kullanabilir. Benzer şekilde e-ticaret siteleri, ürün açıklamalarını otomatik olarak yaratabilir veya görselleri çeşitlendirebilir.

Üretken yapay zekâda dikkat edilmesi gereken bir diğer konu, etik ve güvenliktir. Üretilen içerikler yanlış bilgilendirme, telif hakkı ihlalleri veya uygunsuz içerik riski taşıyabilir. Bu nedenle sistemler, filtreleme mekanizmaları ve denetim katmanlarıyla desteklenir. Örneğin bir haber ajansı, modelin ürettiği metni yayınlamadan önce doğruluk ve bağlam açısından kontrol eder.

Özetle, üretken yapay zekâ; metin, görüntü, ses ve diğer medya türlerinde tamamen yeni içerikler oluşturabilen sistemleri kapsar. DALL-E ve Midjourney gibi görsel üretken modeller, sanat ve tasarım alanında ilham kaynağı olurken; GPT-4, ChatGPT ve benzeri büyük dil modelleri metin üretiminde devrim yaratmaktadır. Tıp, eğitim, oyun ve pazarlama gibi alanlarda üretken yapay zekâ, hem yaratıcı hem de işlevsel çözümler sunar. Python ve açık kaynak kütüphaneler sayesinde kullanıcılar, bu sistemlerin nasıl çalıştığını deneyimleyebilir ve kendi projelerinde uygulayabilir. Üretken yapay zekânın gücü, özgün içerik yaratma kapasitesi ve insan benzeri yaratıcılığı simüle etme yeteneğinde yatmaktadır ve gelecekte pek çok sektörde daha da kritik bir rol üstleneceği öngörülmektedir.

Büyük Dil Modelleri: Dilin Devleri

Büyük dil modelleri, milyarlarca kelime ve cümle üzerinden eğitilmiş yapay zekâ sistemleridir. GPT-4, LLaMA, Claude ve Mistral gibi modeller, insan dilinin karmaşıklığını anlamak için tasarlanmıştır. Bu modeller, dilin kurallarını, kelime dizilimlerini, bağlam ilişkilerini ve anlamsal ipuçlarını öğrenir. Modelin eğitimi, tıpkı bir öğrencinin kitaplardan ve deneyimlerden öğrenmesi gibi düşünülebilir; ne kadar çok metin okursa, bağlamı o kadar iyi kavrar ve anlamlı cümleler üretebilir.

Bir büyük dil modeli, kelimelerin birbirini takip etme olasılıklarını hesaplar. Örneğin "Sabah kahvemi içerken güne başlamak…" cümlesinin devamını tahmin etmek için model, önceki kelimeleri ve olası bağlamları analiz eder. Sonuç olarak "Sabah kahvemi içerken güne başlamak bana enerji veriyor ve üretken olmamı sağlıyor." gibi doğal bir cümle ortaya çıkar. Bu, basit bir kelime tahmininden çok daha fazlasıdır; model bağlamı, anlamı ve dilin inceliklerini kullanır.

Büyük dil modelleri yalnızca metinle sınırlı kalmaz. DALL-E ve Midjourney gibi modeller, kullanıcıların verdiği açıklamalara uygun görseller üretebilir. Örneğin "uzayda kahve içen bir robot" dediğinizde, sistem size tamamen özgün ve hayal ürünü bir resim sunar. OpenAI Jukebox veya Google MusicLM gibi modeller ise metin tabanlı açıklamalardan müzik yaratabilir. Bu çeşitlilik, üretken yapay zekâyı yalnızca bir araç değil, yaratıcı bir iş ortağı hâline getirir.

ChatGPT'nin Basitleştirilmiş Çalışma Mantığı

ChatGPT, GPT-4 üzerine inşa edilmiş bir sohbet modelidir. Basitleştirilmiş olarak, modelin işleyişi üç aşamada özetlenebilir:

1. **Girdi Alma:** Kullanıcı, modelle etkileşime geçer ve bir mesaj yazar. Örneğin: "Bana kısa bir bilim kurgu hikâyesi yaz."
2. **Bağlam Analizi:** Model, hem kullanıcı mesajını hem de önceki konuşmaları dikkate alır. Böylece bağlamı koruyarak anlamlı bir yanıt üretir.
3. **Olasılık Hesabı ve Metin Üretimi:** Model, hangi kelimenin veya cümlenin devamında gelme olasılığı en yüksekse onu seçer ve metni oluşturur. Bu işlem, metin tamamlanana kadar devam eder.

Örnek: Kullanıcı "Ortaçağda geçen bir bilim kurgu hikâyesi yaz" dediğinde model şu şekilde yanıt verebilir: "Krallığın gölgesinde, yıldızlararası bir yolculuğa çıkan genç bir mucit vardı. Gökyüzündeki devasa makineler, köyün taş sokaklarında yankılanıyordu. Bu mucit, zamanı bükmenin sırlarını keşfetti ve kaderini değiştirdi." Model burada bağlamı, olay örgüsünü ve karakterleri dikkate almıştır; bu da büyük dil modellerinin sadece kelime üretmekten öte bağlam ve anlam üzerine kurulduğunu gösterir.

Metin, Görsel ve Müzik Üretimi

Üretken yapay zekâ, yaratıcı alanlarda üç ana biçimde karşımıza çıkar:

1. **Metin Üretimi:** ChatGPT ve GPT-4, hikâye, şiir, makale veya teknik metinler üretebilir. Örnek olarak bir öğrenci "Mars'ta geçen bir dedektif hikâyesi yaz" dediğinde model, gezegenin atmosferini, karakterleri ve olay örgüsünü özgün biçimde birleştirir. Bu metinler, yalnızca eğlence için değil, eğitim ve iş dünyasında da kullanılabilir. Örneğin, bir öğretmen ders notlarını özetlemek veya öğrenciler için hikâyeler oluşturmak için modeli kullanabilir.
2. **Görsel Üretimi:** DALL-E, Midjourney ve Stable Diffusion, metin açıklamalarına uygun özgün görseller yaratır. Örnek: "Buzullar arasında dans eden flamingolar" cümlesi verildiğinde sistem, daha önce görülmemiş ve yaratıcı bir görsel üretir. Bu görseller, reklam, sosyal medya içerikleri, kitap kapakları veya oyun tasarımı gibi alanlarda kullanılabilir.
3. **Müzik Üretimi:** OpenAI Jukebox, MusicLM veya AIVA gibi modeller, metin veya kısa örnekler üzerinden müzik üretebilir. Örneğin, kullanıcı "Rahatlatıcı bir klasik piyano parçası" dediğinde model özgün bir piyano müziği yaratır. Bu tür müzikler, film, oyun veya meditasyon uygulamalarında kullanılabilir. Üstelik kullanıcı, müziğin temposunu, enstrümanlarını veya tarzını da belirleyebili

Halüsinasyon Problemi

Üretken yapay zekânın en büyük sınırlılıklarından biri halüsinasyon problemidir. Halüsinasyon, modelin yanlış veya hayali bilgiler üretmesidir.

Örnek:

- Kullanıcı: “Dünyadaki en yüksek dağ nedir?”
- Model: “Mount Olympus, 10.000 metre yüksekliğinde.”
- Gerçek: Dünyanın en yüksek dağı Everest’tir (8.848 metre).

Halüsinasyonun temel sebepleri şunlardır:

- Eğitim verilerinde hatalı veya eksik bilgiler bulunması
- Modelin olasılık tahmini yaparken kesinlik yerine olasılığa dayanması
- Gerçek zamanlı doğrulama yeteneğinin olmaması

Çözüm yolları arasında, insan denetimi, güvenilir veri kaynaklarıyla güncel model eğitimi ve yanıtların doğruluk güven aralıklarıyla verilmesi yer alır.

Prompt Yazma Sanatı

Prompt, üretken yapay zekâya verilen komut veya talimattır. Prompt’un kalitesi, modelin çıktısının kalitesini doğrudan etkiler. İyi bir prompt, net, detaylı ve bağlamlı olmalıdır.

Örnekler:

- **Kısa ve belirsiz prompt:** “Hikâye yaz.” → Genel ve bağlamdan yoksun bir metin elde edilir.
- **Ayrıntılı prompt:** “Ortaçağda geçen, uzaylıların ziyaret ettiği bir köyde yaşayan genç bir mucidin macerasını anlatan, 3 paragraf uzunluğunda bir hikâye yaz.” → Model, karakter, mekan ve olay örgüsü içeren tutarlı bir hikâye üretir.
- **Soru tabanlı prompt:** “Mars’ta kolonileşme ile ilgili 5 kısa madde yaz.” → Net ve odaklı bir çıktı sağlar.
- **Yaratıcı prompt örnekleri:**
    - “Rönesans tarzında bir resim olarak, kedi ve robotun birlikte çay içtiği sahneyi tarif et.”
    - “Lo-fi hip hop tarzında, yağmurlu bir İstanbul sabahını anlatan 30 saniyelik müzik üret.”
    - “Şiir şeklinde, yapay zekâ ve insan iş birliğini anlatan 4 kıtalık kısa bir şiir yaz.”
- **Problem çözme promptları:**
    - “Python kullanarak basit bir hesap makinesi uygulamasının kodunu yaz.”
    - “Bir lise öğrencisi için fotosentez konusunu 5 maddede açıkla.”

Prompt yazarken dikkat edilmesi gerekenler:

- Bağlam vermek: Modelin ne hakkında yazacağını netleştirmek
- Detay ve uzunluk: Kısa ve belirsiz prompt'lar genellikle başarısız olur
- Format belirtmek: Madde listesi, paragraf, kod veya şiir
- Hedef kitleyi belirtmek: Öğrenci, yetişkin, uzman vb.

Eğitimde, öğrenciler ders özetleri veya hikâyeler için prompt kullanabilir. İş dünyasında rapor, e-posta taslağı veya sunum metinleri üretmek için kullanılabilir. Yaratıcı alanlarda ise hikâye, şiir, görsel ve müzik üretiminde etkilidir. Programlama alanında ise kod üretimi, hata ayıklama ve algoritma açıklamaları sağlanabilir.

Örnek:
Prompt: "Bir startup kuracak öğrencilere, yapay zekâ destekli bir uygulama fikri ve 3 adımlık iş planı öner."
Model yanıtı:

1. Uygulama Fikri: "Akıllı Öğrenme Asistanı" – Öğrencilerin ders çalışma alışkanlıklarını analiz eden ve kişiselleştirilmiş çalışma planı sunan bir uygulama.
2. Adım 1: Pazar araştırması ve hedef kullanıcı analizi.
3. Adım 2: Prototip geliştirme ve test kullanıcıları ile deneme.
4. Adım 3: Lansman ve geri bildirim toplama.

Bu örnek, prompt'un netliği ve detayının modelin çıktısını doğrudan etkilediğini gösterir.

Bölüm Özeti

Bu bölümde, yapay zekânın en hızlı gelişen ve en yaratıcı alanlarından biri olan üretken yapay zekâ (Generative AI) ele alınmıştır. Üretken yapay zekâ, yalnızca verileri analiz eden sistemlerden farklı olarak, öğrenilen bilgilerden tamamen yeni içerikler üretebilen modellerdir. Metin, görsel, müzik ve hatta video gibi farklı türlerde içerik üretme yeteneği sayesinde, bu sistemler adeta bir yazar, sanatçı veya besteci gibi davranabilir. Ancak bu "yaratıcılık", insan düşüncesinden farklı olarak olasılık hesapları ve istatistiksel örüntüler üzerine kuruludur.

Bölümde, üretken yapay zekânın hem metin hem de görsel üretim alanındaki uygulamaları incelenmiştir. DALL-E ve Midjourney gibi modeller, metin açıklamalarından özgün görseller üretirken; GPT-4 ve benzeri büyük dil modelleri (LLM), metin üretimi, özetleme, çeviri ve soru-cevap gibi görevlerde kullanılmaktadır. Python üzerinden verilen GPT-2 örneği ile, bir metnin nasıl token'lara ayrıldığı, modelin bu token'lar üzerinden nasıl tahmin yaptığı ve yeni metinler ürettiği adım adım açıklanmıştır.

Üretken yapay zekânın kullanım alanlarının yalnızca sanatla sınırlı olmadığı; eğitim, tıp, mühendislik, oyun geliştirme ve pazarlama gibi birçok alanda önemli katkılar sağladığı vurgulanmıştır. Örneğin tıp alanında ilaç tasarımı, eğitimde içerik üretimi ve oyun sektöründe karakter diyalogları bu teknolojinin kullanımına örnek olarak verilmiştir.

Büyük dil modelleri (LLM), milyarlarca kelime üzerinde eğitilerek dilin yapısını, bağlamını ve anlam ilişkilerini öğrenir. Bu modeller, bir cümlenin devamını tahmin ederken yalnızca kelimeleri değil, bağlamı da dikkate alır. ChatGPT gibi sistemlerin çalışma mantığı; girdi alma, bağlam analizi ve olasılık hesaplayarak metin üretme adımlarından oluşur.

Bölümde ayrıca üretken yapay zekânın önemli bir sorunu olan halüsinasyon problemi ele alınmıştır. Bu problem, modelin gerçekte doğru olmayan bilgileri üretmesi anlamına gelir ve genellikle eksik veri, olasılıksal tahmin yapısı ve doğrulama eksikliğinden kaynaklanır.

Son olarak, üretken yapay zekâdan etkili sonuçlar alabilmek için prompt yazma sanatı üzerinde durulmuştur. İyi bir prompt'un açık, detaylı ve bağlam içeren bir yapıya sahip olması gerektiği vurgulanmıştır. Doğru yazılmış bir prompt, modelin daha tutarlı, yaratıcı ve hedefe uygun çıktılar üretmesini sağlar.

Genel olarak bu bölüm, üretken yapay zekânın nasıl çalıştığını, nerelerde kullanıldığını, güçlü ve zayıf yönlerini ve bu sistemlerle etkili iletişim kurmanın yollarını kapsamlı bir şekilde açıklamaktadır.

Soru-Cevap

1. Üretken yapay zekâ (Generative AI) nedir?
Üretken yapay zekâ, mevcut verilerden öğrenerek yeni ve özgün içerikler (metin, görsel, müzik vb.) üretebilen yapay zekâ sistemleridir.

2. Üretken yapay zekâ ile klasik yapay zekâ arasındaki fark nedir?
Klasik yapay zekâ analiz ve sınıflandırma yaparken, üretken yapay zekâ yeni içerikler oluşturabilir.

3. Büyük dil modelleri (LLM) nedir?
Büyük dil modelleri, milyarlarca kelime üzerinde eğitilmiş ve insan dilini anlayıp anlamlı metinler üretebilen yapay zekâ sistemleridir.

4. GPT modelleri nasıl çalışır?
Verilen metni analiz eder, kelimelerin olasılıklarını hesaplar ve en uygun kelimeleri seçerek metni devam ettirir.

5. DALL-E ve Midjourney ne tür işler yapar?
Bu modeller, metin açıklamalarına dayanarak özgün ve yaratıcı görseller üretir.

6. Üretken yapay zekâ hangi alanlarda kullanılır?
Eğitim, tıp, sanat, oyun geliştirme, pazarlama ve yazılım geliştirme gibi birçok alanda kullanılır.

7. Halüsinasyon problemi nedir?
Modelin yanlış veya gerçekte olmayan bilgileri doğruymuş gibi üretmesidir.

8. Halüsinasyon neden oluşur?
Modelin olasılıklara dayalı çalışması, eksik veya hatalı eğitim verileri ve doğrulama mekanizmasının olmaması nedeniyle oluşur.

9. Prompt nedir?
Prompt, yapay zekâya verilen komut veya yönlendirmedir ve modelin üreteceği çıktıyı doğrudan etkiler.

10. İyi bir prompt nasıl olmalıdır?
Açık, detaylı, bağlam içeren ve hedefi net belirten bir şekilde yazılmalıdır.

GELECEK VE İNSAN–MAKİNE İLİŞKİSİ

Yapay zekâ, günümüzde yalnızca bilim kurgu filmlerinin bir parçası değil; yaşamımızın hemen her alanında var olan, etkisi giderek artan bir teknoloji hâline geldi. Ancak teknoloji geliştikçe, özellikle "insan ile makine arasındaki ilişki" konusu daha fazla tartışılır hâle geliyor. Bu bölümde, süper zekâ tartışmalarından başlayarak geleceğin mesleklerini, insan ve yapay zekâ işbirliğini ve öğrenciler için olası yol haritalarını detaylı şekilde inceleyeceğiz.

Süper Zekâ Tartışmaları

Süper zekâ, insan zekâsını aşan, kendi kendine öğrenme ve kendi amaçlarını belirleme yeteneğine sahip yapay zekâ sistemleri olarak tanımlanır. Elon Musk, Nick Bostrom ve Ray Kurzweil gibi önde gelen düşünürler, süper zekânın hem büyük fırsatlar hem de ciddi riskler doğurabileceğini öne sürmektedir. Musk, süper zekânın kontrolsüz gelişmesi hâlinde insanlık için varoluşsal bir risk oluşturabileceğini savunurken, Bostrom "Büyük Filtre" yaklaşımıyla bu riskin stratejik yönetilmesi gerektiğini vurgular.

Ancak şu anda elimizdeki tüm yapay zekâ sistemleri hâlâ "dar yapay zekâ" seviyesinde. Yani satrançta dünya şampiyonunu yenebilen bir sistem, başka bir alanda hiçbir yetenek gösteremez. Örneğin AlphaGo, Go oyununda insanları alt etti, ama aynı algoritma bir şiiri analiz edemez veya bir arabayı güvenli şekilde süremez. Dolayısıyla süper zekâ hâlâ teorik bir tartışmadır, fakat bu tartışma, geleceğe hazırlıklı olmanın önemini ortaya koyar.

Süper zekâ tartışmalarının en önemli boyutu etik ve güvenliktir. Eğer bir sistem kendi hedeflerini belirleyebiliyorsa ve bu hedefler insan çıkarlarıyla çelişiyorsa ne olur? Örneğin, bir süper zekâ kaynak kullanımını optimize ederken insanlar için gerekli olan su ve enerji kaynaklarını göz ardı edebilir. Bu nedenle araştırmacılar, süper zekâ sistemlerinin güvenliğini sağlayacak kurallar ve sınırlar üzerinde çalışmaktadır.

Geleceğin Meslekleri

Yapay zekânın gelişimi, iş dünyasında da önemli değişiklikler yaratmaktadır. Bazı meslekler yok olurken, yeni meslekler ortaya çıkacaktır. Geleceğin mesleklerini anlamak, öğrenciler ve yeni mezunlar için kritik önemdedir.

Günümüzde veri analistleri, makine öğrenmesi mühendisleri ve yapay zekâ etik uzmanları gibi meslekler hızla önem kazanmaktadır. Gelecekte bu mesleklerin yanı sıra "AI etkileşim tasarımcıları", "sanal eğitim mentorları" ve "yapay zekâ güvenlik denetçileri" gibi yeni

pozisyonlar doğacaktır. Örneğin bir fabrika, üretim hattında tamamen otomatik bir sistem kullanabilir; bu sistemin verimli ve güvenli çalışmasını sağlamak için mühendisler ve denetçiler, makinelerle sürekli etkileşim hâlinde olacaktır.

Bazı meslekler ise daha yaratıcı ve insana özgü yetenekler gerektirecek şekilde yeniden şekillenecektir. Örneğin bir reklam ajansı, yapay zekâ ile tasarım fikirleri üretebilir, ancak insan yaratıcılığı, bağlam ve kültürel nüansları yönetmek için hâlâ vazgeçilmez olacaktır. Benzer şekilde bir doktor, yapay zekânın sunduğu analizleri kullanarak daha hızlı ve doğru teşhis koyabilir, ancak hasta ile empati kurma ve kararları insan değerleriyle dengeleme yeteneği sadece insana özgüdür.

İnsan ve AI İşbirliği

Gelecekteki en önemli trend, insan ve yapay zekâ işbirliği olacaktır. İnsanlar, yapay zekânın hızlı hesaplama, büyük veri analizi ve örüntü tanıma yeteneklerinden faydalanacak, yapay zekâ ise insanın yaratıcı ve sosyal zekâ özelliklerinden güç alacaktır.

Örnek olarak sağlık alanını ele alalım. Yapay zekâ, röntgen ve MR görüntülerini analiz ederek olası tümörleri yüksek doğrulukla tespit edebilir. Ancak hasta ile iletişim kurmak, tedavi seçeneklerini tartışmak ve duygusal destek vermek hâlâ insan doktorun görevidir. Bu işbirliği, hem teşhis hem de tedavi süreçlerini daha etkili hâle getirir.

Eğitim alanında da benzer bir işbirliği görülebilir. Öğrenciye yönelik kişiselleştirilmiş öğrenme planları, yapay zekâ ile hazırlanabilir; öğrencinin güçlü ve zayıf yönleri anında analiz edilir ve öneriler sunulur. Ancak rehberlik, motivasyon ve yaratıcı problem çözme hâlâ öğretmenin rolüdür. Bu durum, “AI ile öğrenmek” ve “AI’ye bağımlı olmak” arasındaki çizginin önemini gösterir. Eğer öğrenciler sadece yapay zekânın sunduğu cevaplara güvenirse, eleştirel düşünme ve bağımsız öğrenme becerileri zayıflar.

İş dünyasında da insan + AI işbirliği büyük bir avantaj sağlar. Örneğin bir finansal analizci, yapay zekânın sağladığı büyük veri tahminlerini kullanarak yatırım stratejileri geliştirebilir. Yapay zekâ, risk analizi ve geçmiş verilerin değerlendirilmesinde hızlıdır, ancak nihai karar ve strateji belirleme hâlâ insanın deneyimi ve sezgisi ile gerçekleşir.

Öğrenciler İçin Yol Haritası

Öğrenciler için insan ve yapay zekâ işbirliğini anlamak, geleceğe hazırlanmak açısından kritik önemdedir. İlk adım, temel yapay zekâ kavramlarını öğrenmek ve bunları günlük hayatta gözlemlemektir. Örneğin bir öğrenci, sosyal medyadaki içerik önerilerini veya çevrim içi alışveriş önerilerini analiz ederek makine öğrenmesinin mantığını gözlemleyebilir.

Bir sonraki adım, basit kodlama ve veri analizi becerilerini kazanmaktır. Python veya Scratch gibi araçlarla küçük makine öğrenmesi projeleri yapmak, öğrencinin algoritmaların nasıl çalıştığını anlamasını sağlar. Örneğin, bir öğrenci basit bir e-posta spam sınıflandırma modeli veya el yazısı rakam tanıma projesi yapabilir.

Daha sonra öğrenciler, yapay zekâ ile işbirliği projelerine katılabilir. Örneğin bir okul robot kulübü, öğrencilerin küçük robotları programlamasına ve yapay zekâ destekli görevler yapmasına olanak tanır. Bu tür projeler, hem teknik becerileri hem de yaratıcı problem çözme yeteneklerini geliştirir.

Son olarak, etik ve eleştirel düşünceyi öğrenmek gerekir. Yapay zekânın kararları her zaman doğru olmayabilir; veri önyargısı veya yanlış algoritmalar hatalı sonuçlar doğurabilir. Öğrencilerin, makine çıktısını sorgulama, analiz etme ve kendi mantıklarıyla doğrulama becerilerini geliştirmesi, gelecekteki iş hayatı ve akademik başarı açısından büyük avantaj sağlar.

Geleceğe Dair Öngörüler

İnsan ve makine ilişkisi gelecekte daha da derinleşecektir. Evlerimizde, işyerlerimizde ve şehirlerimizde yapay zekâ sistemleri günlük hayatın vazgeçilmez bir parçası olacak. Otonom araçlar, akıllı şehir altyapıları, kişiselleştirilmiş eğitim sistemleri ve sağlık analiz platformları, insan yaşamını hem daha güvenli hem de daha verimli hâle getirecek.

Buna karşılık, süper zekâ tartışmaları hâlâ gündemde olacak. Toplumun, iş dünyasının ve eğitim sistemlerinin, yapay zekâyı bilinçli ve etik şekilde kullanmayı öğrenmesi gerekiyor. İnsan ve makine arasındaki ilişkiyi yalnızca teknolojik bir konu olarak değil, aynı zamanda sosyal, etik ve ekonomik bir mesele olarak görmek önemlidir.

Özetle, gelecek insan ve yapay zekânın birlikte çalışabildiği bir dünya vaat ediyor. İnsanlar yaratıcı düşünme, empati, etik kararlar ve stratejik planlama gibi yetenekleri korurken, yapay zekâ sistemleri veri analizi, hızlı hesaplama ve örüntü tanıma gibi görevlerde destek sağlayacak. Bu işbirliği, eğitim, sağlık, iş ve günlük yaşamda yeni fırsatlar ve meydan okumalar yaratacak.

Öğrenciler ve gençler için kritik olan, yapay zekâyı yalnızca bir araç olarak görmek ve ona bağımlı olmadan, eleştirel bir bakış açısıyla kullanmayı öğrenmektir. Bu yaklaşım, gelecekte hem bireysel başarı hem de toplumsal fayda için hayati öneme sahiptir.

İlerleyen yıllarda, insan–makine etkileşimi daha da gelişecek ve bugünkü tartışmalar, gelecekteki iş dünyası ve yaşam biçimi için bir temel oluşturacaktır. Bu nedenle öğrencilerin, araştırmaya açık, meraklı ve sorumlu bir şekilde yapay zekâ ile etkileşimde bulunmaları, hem kendi kariyerlerini hem de toplumu daha iyi bir noktaya taşıyacaktır.

Bölüm Özeti

Bu bölüm, yapay zekânın günümüzde ve gelecekte insan yaşamındaki rolünü, özellikle insan–makine ilişkisini ele almaktadır. Yapay zekâ yalnızca bilim kurgu konsepti olmaktan çıkarak günlük yaşamın birçok alanına entegre olmuştur. Ancak teknoloji geliştikçe, süper zekâ ve insan–makine işbirliği gibi konular tartışma konusu olmaktadır.

Süper Zekâ Tartışmaları: Süper zekâ, insan zekâsını aşan ve kendi hedeflerini belirleyebilen yapay zekâ sistemleri olarak tanımlanır. Elon Musk, Nick Bostrom ve Ray Kurzweil gibi

düşünürler, süper zekânın hem fırsatlar hem de riskler barındırdığını vurgular. Günümüzdeki sistemler hâlâ dar yapay zekâ seviyesindedir; örneğin AlphaGo yalnızca Go oyununda yeteneklidir. Süper zekâ tartışmaları etik ve güvenlik boyutlarıyla önemlidir; kontrolsüz bir sistem insan çıkarlarıyla çelişebilir. Bu nedenle kurallar ve sınırlar üzerinde çalışmalar sürmektedir.

Geleceğin Meslekleri: Yapay zekânın gelişimi bazı meslekleri dönüştürürken, yeni meslekler de ortaya çıkarmaktadır. Veri analistleri, makine öğrenmesi mühendisleri ve yapay zekâ etik uzmanları günümüzde öne çıkan mesleklerdir. Gelecekte "AI etkileşim tasarımcıları", "sanal eğitim mentorları" ve "yapay zekâ güvenlik denetçileri" gibi yeni pozisyonlar önem kazanacaktır. Ayrıca bazı meslekler, insan yaratıcılığı ve kültürel bağlamı gerektiren yeteneklerle yeniden şekillenecektir.

İnsan ve AI İşbirliği: Gelecekte insan ve yapay zekâ işbirliği temel bir trend olacaktır. İnsanlar yaratıcı ve sosyal zekâ yeteneklerini kullanırken, yapay zekâ büyük veri analizi, hızlı hesaplama ve örüntü tanıma gibi görevlerde destek sağlayacaktır. Sağlık, eğitim ve iş dünyasında bu işbirliği verimliliği artırır, ancak eleştirel düşünme ve insan karar yetenekleri hâlâ önemlidir. Örneğin, doktorlar yapay zekâ analizlerini kullanarak teşhis koyabilir, fakat hasta ile empati kurma ve karar dengeleme hâlâ insana aittir.

Öğrenciler İçin Yol Haritası: Öğrencilerin geleceğe hazırlanması için temel yapay zekâ kavramlarını öğrenmeleri, kodlama ve veri analizi becerileri kazanmaları ve yapay zekâ ile işbirliği projelerine katılmaları önerilmektedir. Ayrıca etik ve eleştirel düşünme becerileri geliştirmek, model çıktılarının doğruluğunu sorgulamak önemlidir.

Geleceğe Dair Öngörüler: İnsan–makine ilişkisi ilerleyen yıllarda daha da derinleşecektir. Otonom araçlar, akıllı şehirler ve kişiselleştirilmiş eğitim ve sağlık sistemleri hayatın vazgeçilmez parçaları olacaktır. Süper zekâ tartışmaları sürecek ve toplumsal, etik ve ekonomik boyutlarıyla ele alınacaktır. İnsanlar yaratıcı düşünme, empati ve etik kararlar gibi benzersiz yetenekleri korurken, yapay zekâ veri analizi ve hızlı hesaplama ile destek sağlayacaktır. Öğrenciler ve gençler için kritik olan, yapay zekâyı bir araç olarak görmek ve ona bağımlı olmadan eleştirel kullanmayı öğrenmektir.

Özetle, gelecek insan ve yapay zekânın birlikte çalışabildiği bir dünya vaat etmekte; bu işbirliği eğitimden sağlığa, iş dünyasından günlük yaşama kadar yeni fırsatlar ve meydan okumalar sunmaktadır.

Soru-Cevap

1. **Soru: Süper zekâ nedir ve hangi yeteneklere sahiptir?**
   Cevap: Süper zekâ, insan zekâsını aşan, kendi kendine öğrenme ve kendi amaçlarını belirleyebilme yeteneğine sahip yapay zekâ sistemleridir.

2. **Soru: Elon Musk ve Nick Bostrom süper zekâ konusunda hangi riskleri vurgular?**
   Cevap: Musk, kontrolsüz süper zekânın insanlık için varoluşsal risk oluşturabileceğini, Bostrom ise bu riskin stratejik yönetilmesi gerektiğini belirtir.

3. **Soru: Günümüzdeki yapay zekâ sistemlerinin kapasitesi nasıldır?**
   Cevap: Şu anda elimizdeki yapay zekâ sistemleri "dar yapay zekâ" seviyesindedir; yani belirli bir alanda çok başarılı olabilirler, fakat başka alanlarda yetenek gösteremezler.

4. **Soru: Süper zekâ tartışmalarında etik ve güvenlik neden önemlidir?**
   Cevap: Çünkü süper zekâ kendi hedeflerini belirleyebiliyorsa, bu hedefler insan çıkarlarıyla çelişebilir ve kaynak kullanımında ya da karar almada insanları olumsuz etkileyebilir.

5. **Soru: Yapay zekâ gelecekte meslekleri nasıl değiştirecek?**
   Cevap: Bazı meslekler yok olurken, yeni meslekler ortaya çıkacak ve bazı meslekler yaratıcı ve insana özgü yetenekler gerektirecek şekilde yeniden şekillenecek.

6. **Soru: Gelecekte öne çıkması beklenen yapay zekâ odaklı meslekler nelerdir?**
   Cevap: AI etkileşim tasarımcıları, sanal eğitim mentorları, yapay zekâ güvenlik denetçileri, makine öğrenmesi mühendisleri ve yapay zekâ etik uzmanları gibi meslekler öne çıkacaktır.

7. **Soru: İnsan ve yapay zekâ işbirliği hangi alanlarda avantaj sağlar?**
   Cevap: Sağlıkta teşhis ve tedavi süreçlerinde, eğitimde kişiselleştirilmiş öğrenmede ve iş dünyasında veri analizi ve strateji geliştirmede insan + AI işbirliği büyük avantaj sağlar.

8. **Soru: Öğrenciler yapay zekâ ile çalışmaya nasıl hazırlanabilir?**
   Cevap: Temel yapay zekâ kavramlarını öğrenerek, basit kodlama ve veri analizi becerileri kazanarak, AI destekli projelere katılarak ve etik-eleştirel düşünceyi öğrenerek hazırlanabilirler.

9. **Soru: Yapay zekâ eğitimi öğrenciler için neden kritik önemdedir?**
   Cevap: Çünkü yapay zekâ yalnızca bir araçtır; öğrenciler eleştirel düşünme ve bağımsız öğrenme becerilerini geliştirerek, AI'ye bağımlı olmadan yetkin bir şekilde kullanmayı öğrenmelidir.

10. **Soru: Gelecekte insan–makine ilişkisi nasıl bir dönüşüm yaşayacak?**
    Cevap: İnsan ve makine ilişkisi daha derinleşecek, evlerimizde, işyerlerimizde ve şehirlerimizde AI sistemleri günlük yaşamın vazgeçilmez bir parçası olacak; bu işbirliği yeni fırsatlar ve meydan okumalar yaratacaktır.

TEACHABLE MACHINE (GOOGLE) – MAKİNEYİ ÖĞRETMEK ARTIK KOLAY

Geleneksel makine öğrenmesi, genellikle programcıların karmaşık kodlar yazmasını ve büyük veri setleri hazırlamasını gerektirir. Ancak Google, *Teachable Machine* adını verdiği bir platformla bu süreci herkes için erişilebilir hâle getirdi. Artık öğrenciler, öğretmenler veya meraklı kullanıcılar, herhangi bir yazılım bilgisi olmadan kendi makine öğrenmesi modellerini oluşturabilirler.

Teachable Machine'in temel fikri çok basittir: *"Makineyi örneklerle öğret."* Siz sisteme örnek veriler gösteriyorsunuz, makine ise bu örnekleri öğrenip yeni veriler için tahmin yapabiliyor.

Teachable Machine, *makine öğrenmesi kavramlarını deneyimleyerek öğrenmek* için mükemmel bir araçtır. Kullanıcılar şu temel kavramları kolayca görebilir:

- **Veri toplama ve etiketleme:** Modelin öğrenebilmesi için örnek veriler gerekir.
- **Model eğitimi:** Makine örnekleri analiz ederek sınıflandırmayı öğrenir.
- **Tahmin ve doğruluk:** Modelin ne kadar doğru çalıştığını test edebiliriz.
- **Hata ve düzeltme:** Yanlış tahminler modeli geliştirmek için geri bildirim sağlar.

Örnek: Bir sınıf projesinde öğrenciler, Teachable Machine ile farklı bitki yapraklarını tanıtan bir model oluşturabilir. Her yaprak türü için birkaç fotoğraf yükler, modeli eğitir ve sınıflandırmanın doğruluğunu test eder. Böylece teorik makine öğrenmesi kavramlarını pratiğe dönüştürmüş olurlar.

Google Teachable Machine, karmaşık kodlar yazmadan sadece tarayıcınızı kullanarak kendi yapay zeka modellerinizi eğitmenize olanak tanıyan harika bir araçtır. Lise seviyesinde hem eğlenceli hem de öğretici olabilecek üç farklı proje örneğini aşağıda bulabilirsin:

Duruş ve Hareket Kontrolü (Görüntü Tanıma)

Bu projede bilgisayarın senin farklı hareketlerini tanımasını sağlayabilirsin.

- **Senaryo:** Bilgisayarın senin "Düzgün Oturma" ve "Kambur Oturma" arasındaki farkı anlamasını sağla.
- **Nasıl Yapılır?**
    1. **Sınıf 1 (Dik):** Kameranın karşısında dik dururken 30-40 fotoğraf çek.
    2. **Sınıf 2 (Kambur):** Bilgisayar başında eğilmiş veya yanlış pozisyondayken fotoğraf çek.
- **Sonuç:** Modelini eğittikten sonra, kambur durduğunda bilgisayarının seni uyaran bir ses çıkarmasını (veya ekranda "Lütfen dik otur!" yazmasını) sağlayacak bir sistemin temelini atmış olursun.

Atık Ayrıştırma Rehberi (Nesne Tanıma)

Çevre bilinci oluşturmak için harika bir sınıf projesidir.

- **Senaryo:** Elindeki atığın plastik mi, kağıt mı yoksa metal mi olduğunu söyleyen bir sistem.
- **Nasıl Yapılır?**
    1. **Sınıf 1 (Plastik):** Farklı pet şişeleri ve plastik paketleri kameraya göstererek veri topla.
    2. **Sınıf 2 (Kağıt):** Gazete, defter ve karton parçalarını göster.
    3. **Sınıf 3 (Metal):** İçecek kutularını veya konserveleri göster.
- **Sonuç:** Kamera önüne bir nesne koyduğunda, yapay zekanın bunun hangi geri dönüşüm kutusuna gitmesi gerektiğini anında tahmin etmesini izleyebilirsin.

Sesli Komutla Kontrol (Ses Tanıma)

Görüntü yerine sadece ses kullanarak bir sistemi yönlendirebilirsin.

- **Senaryo:** "Lamba Aç" ve "Lamba Kapat" komutlarını ayırt eden bir model.
- **Nasıl Yapılır?**
    1. **Arka Plan Gürültüsü:** Önce odadaki sessizliği kaydederek modelin "boşluğu" tanımasını sağla.
    2. **Sınıf 1:** "Aç" kelimesini farklı tonlarda defalarca söyleyerek kaydet.
    3. **Sınıf 2:** "Kapat" kelimesini kaydede.
- **Sonuç:** Bu modeli kullanarak, ileride Arduino gibi kartlarla birleştirip gerçekten odanın ışığını kontrol edebilecek bir mantık kurabilirsin.

Teachable Machine Nasıl Çalışır? (Basit Akış)

Yapay zeka eğitim süreci şu üç temel adımdan oluşur:

1. **Veri Toplama (Gather):** Örnekleri (resim, ses veya duruş) sınıflara ayırarak sisteme tanıtırsın.
2. **Eğitme (Train):** Bilgisayar, verdiğin örnekler arasındaki benzerlikleri ve farkları analiz eder.
3. **Test Etme (Test):** Yeni bir nesne veya ses göstererek modelinin ne kadar doğru tahmin yaptığını kontrol edersin.

Teachable Machine projeleri arasında lise düzeyinde en etkileyici ve sonuçları en net görülen **"Atık Ayrıştırma Rehberi"** (Görüntü Tanıma) projesini adım adım ele alalım. Bu proje, hem veri toplama mantığını hem de yapay zekanın "karar verme" sürecini anlamak için harikadır.

Adım Adım: Atık Ayrıştırma Yapay Zekası

1. Hazırlık ve Veri Toplama (Data Gathering)

Teachable Machine sitesine girip **"Image Project"** seçeneğini seçtikten sonra şu sınıfları (klasörleri) oluştur:

- **Sınıf 1: Plastik**
    - **Veri:** Boş pet şişeler, şampuan kutuları veya yoğurt kaplarını kameraya göster.
    - **İpucu:** Şişeyi farklı açılardan (yan, dik, ters) ve farklı uzaklıklardan tutarak "Hold to Record" butonuna bas. Yaklaşık 100-200 kare fotoğraf çekmek iyidir.
- **Sınıf 2: Kağıt**
    - **Veri:** Eski defter sayfaları, kargo kutuları, gazete kağıtları.
    - **İpucu:** Kağıdı biraz buruşturup göster, düz göster. Yapay zeka kağıdın sadece beyaz bir yüzey olmadığını, dokusunu da öğrenmeli.
- **Sınıf 3: Arka Plan (Boş Oda)**
    - **Neden önemli?** Eğer sadece plastik ve kağıdı tanıtırsan, kamera karşısında bir şey yokken yapay zeka kafasına göre birini %100 seçecektir. Arka planı tanıtarak "şu an elimde bir şey yok" demeyi öğretirsin.

2. Eğitme (Training)

Tüm fotoğrafları çektikten sonra ortadaki **"Train Model"** butonuna bas.

- **Püf Noktası:** Eğitim sırasında tarayıcı sekmesini kapatma veya değiştirme.
- **Neler Oluyor?** Bilgisayar şu an pikselleri analiz ediyor. Örneğin; plastiğin üzerindeki ışık yansımasını (parlaklığı) veya kağıdın mat dokusunu ayırt eden matematiksel bir harita çıkarıyor.

3. Test Etme ve İyileştirme (Preview)

Sağ taraftaki panelde kameran açılacak. Şimdi daha önce sisteme **hiç göstermediğin** yeni bir plastik şişe veya kağıt parçası al ve kameraya tut.

- **Sonuçları Oku:** Alt taraftaki barların (örneğin: Plastik %98) nasıl değiştiğini izle.
- **Hata Varsa:** Eğer kağıdı plastik sanıyorsa, kağıt sınıfına gidip biraz daha farklı örnek ekle ve tekrar eğit.

Teachable Machine ile Basit Bir Deney

Diyelim ki bilgisayara "gülümseme" ve "serious yüz" ifadelerini tanıtmak istiyorsunuz. Teachable Machine ile bu işlemi şöyle yapabilirsiniz:

1. Platforma girip **"Image Project"** (Görüntü Projesi) seçersiniz.
2. İki sınıf oluşturursunuz:
    - **Sınıf 1:** Gülümseme
    - **Sınıf 2:** Serious yüz
3. Her sınıfa webcam'inizle veya yüklediğiniz fotoğraflarla birkaç örnek eklersiniz.
4. "Train Model" (Modeli Eğit) butonuna tıklarsınız. Teachable Machine, verdiğiniz örnekleri analiz ederek bir yapay zekâ modeli oluşturur.
5. Artık kamera karşısında gülümseyip ciddi bir ifade yaptığınızda, makine bunu doğru sınıfa ayırabilir.

Örnek: 20 fotoğrafla eğitilen bir model, yeni bir fotoğrafa baktığında %90 doğrulukla sınıfını tahmin edebilir.

Teachable Machine'in Özellikleri

- **Kolay Kullanım:** Kod yazmanıza gerek yok. Sadece sürükle-bırak ve örnek ekleme yeterli.
- **Çeşitli Veri Türleri:** Görüntü, ses ve poz projeleri oluşturabilirsiniz. Örneğin:
    - Ses verisi ile "el sallama" veya "alkış" gibi sesleri tanıtabilirsiniz.
    - Poz verisi ile el hareketlerini veya duruşları algılayabilirsiniz.
- **Hızlı Eğitim:** Küçük veri setleri ile bile model birkaç saniye içinde eğitilebilir.
- **İnteraktif Test:** Eğitilen modeli anında test edebilirsiniz; hata ve doğruluk oranlarını görebilirsiniz.

Basit Kod Örneği ile Kullanım

Teachable Machine, eğitilen modeli *web tarayıcıda veya Python ile kullanmanıza* izin verir. Örneğin, görüntü sınıflandırması için Python'da şu şekilde entegre edilebilir:

```
import tensorflow as tf
import numpy as np
from PIL import Image

# Eğitilmiş Teachable Machine modelini yükle
model = tf.keras.models.load_model('my_model.h5')

# Test edilecek görüntüyü yükle
img = Image.open('test_photo.jpg').resize((224, 224))
img_array = np.array(img)/255.0
img_array = np.expand_dims(img_array, axis=0)

# Tahmin yap
predictions = model.predict(img_array)
print("Tahmin:", np.argmax(predictions))
```

Bu örnekle, platformda eğitilen modelinizi kendi bilgisayarınızda veya web projenizde kullanabilirsiniz.

Mini Uygulama: Teachable Machine ile Kendi Modelini Oluştur

Bu bölümde, Teachable Machine kullanarak basit bir görüntü sınıflandırma modeli oluşturacağız. Adımlar öğrencilerin kendi bilgisayarında veya tabletinde kolayca takip edebileceği şekilde tasarlandı.

1. Platforma Giriş

1. Tarayıcınızı açın ve şu adresi ziyaret edin: https://teachablemachine.withgoogle.com/
2. **"Get Started"** (Başlayın) butonuna tıklayın.
3. **"Image Project"** (Görüntü Projesi) seçeneğini seçin.

2. Sınıfları Oluşturma

1. Sol üstteki **"Class 1"** ve **"Class 2"** kutularını göreceksiniz.
2. Bu örnekte **Class 1** = "Gülümseme", **Class 2** = "Serious yüz" olacak.
3. Her sınıfa webcam ile fotoğraf çekin veya bilgisayardan yükleyin.
    - Öneri: Her sınıf için en az 10-20 örnek kullanın.

3. Modeli Eğitme

1. Sağ üstteki **"Train Model"** butonuna tıklayın.

2. Eğitim birkaç saniye sürer (veri miktarına bağlı olarak).
3. Eğitim tamamlandığında model artık tahmin yapabilir.

4. Modeli Test Etme

1. Webcam'i açın veya yeni bir fotoğraf yükleyin.
2. Makine, sizin yaptığınız ifadeyi tahmin edecek.
    - Örnek: Kameranın karşısında gülümsediğinizde, "Gülümseme" sınıfını gösterir.
3. Tahminin doğruluğunu ekranda yüzdelik olarak görebilirsiniz.

5. Hata ve Öğrenme Deneyimi

- Eğer model yanlış tahmin ederse, daha fazla örnek ekleyin veya farklı açılardan fotoğraflar çekin.
- Model, yeni verilerle tekrar eğitildiğinde doğruluğu artar.
- Bu süreç, makine öğrenmesinin **"özyinelemeli öğrenme"** ve **"hata ile düzeltme"** mekanizmasını deneyimlemenin bir yoludur.

6. Modeli Kullanma ve Paylaşma

- Eğitilen modeli bilgisayarınıza indirebilir veya web projesinde kullanabilirsiniz.
- Python, JavaScript veya mobil uygulamalara entegre ederek kendi projelerinizi geliştirebilirsiniz.
- Örnek Python entegrasyonu daha önceki bölümde gösterilmişti.

7. Ekstra Deneyim Önerisi

- **Ses sınıflandırma projesi:** "El sallama" ve "Alkış" seslerini öğretin.
- **Poz sınıflandırma projesi:** Farklı el hareketlerini veya duruşları tanıtın.
- Bu deneyler, öğrencilerin makine öğrenmesi kavramlarını farklı veri türlerinde görmesini sağlar.

Öğrenci Kılavuzu: Teachable Machine ile Kendi Modelini Kurma

Amaç

Bu rehber, öğrencilerin kendi yapay zekâ modellerini **görsel, ses veya poz verileri kullanarak** oluşturmasını sağlar. Deneyim yoluyla makine öğrenmesi kavramlarını anlamayı hedefler.

1. Platforma Giriş

Adımlar:

1. Tarayıcınızı açın ve Teachable Machine sitesine gidin.

2. **“Get Started”** (Başlayın) butonuna tıklayın.
3. **“Image Project”** (Görüntü Projesi) seçeneğini seçin.

İpucu: Eğer bilgisayarınızda webcam yoksa, fotoğraf yükleyerek de projeye başlayabilirsiniz.

2. Sınıfları Oluşturma

Adımlar:

1. Sol üstte **“Class 1”** ve **“Class 2”** kutularını göreceksiniz.
2. Örnek proje:
    - **Class 1 = Gülümseme**
    - **Class 2 = Serious yüz**
3. Her sınıfa webcam veya yüklediğiniz fotoğraflarla **en az 10-20 örnek** ekleyin.
4. Örnekleri farklı açılardan ve ışık koşullarında çekmek, modelin doğruluğunu artırır.

İpucu: Çeşitli pozlar ve ifadeler kullanmak, modelin genelleme yeteneğini geliştirir.

3. Modeli Eğitme

Adımlar:

1. Sağ üstteki **“Train Model”** butonuna tıklayın.
2. Eğitim birkaç saniye sürecektir.
3. Eğitim tamamlandığında model, yeni verileri tahmin etmeye hazırdır.

İpucu: Daha fazla örnek yükleyerek veya farklı açılardan fotoğraf çekerek modelin doğruluğunu artırabilirsiniz.

4. Modeli Test Etme

Adımlar:

1. **Webcam’i açın** veya yeni bir fotoğraf yükleyin.
2. Makine, sizin yaptığınız ifadeyi veya yüklediğiniz görseli sınıflandıracak.
    - Örnek: Kamera karşısında gülümseyin → model “Gülümseme” sınıfını gösterir.
3. Tahmin doğruluğu ekranda yüzdelik olarak görünür.

İpucu: Model yanlış tahmin ediyorsa, daha fazla veri ekleyin ve tekrar eğitin.

5. Ses ve Poz Projeleri ile Deney

- **Ses Projesi:**

1. “El sallama” ve “Alkış” gibi sesleri kaydedin.
2. Modeli eğitin ve farklı sesleri doğru sınıfa ayırmasını test edin.

- **Poz Projesi:**
    1. Farklı el hareketlerini veya duruşları gösterin.
    2. Webcam ile modeli eğitin ve doğru sınıf tahminlerini test edin.

İpucu: Ses ve poz projeleri, makine öğrenmesinin sadece görüntü değil, farklı veri türlerinde de uygulanabileceğini gösterir.

6. Modeli Kullanma ve Paylaşma

1. Eğitilen modeli bilgisayarınıza indirin veya web projesinde kullanın.
2. Python veya JavaScript ile entegre edebilirsiniz:

```
import tensorflow as tf
import numpy as np
from PIL import Image

# Eğitilmiş modeli yükle
model = tf.keras.models.load_model('my_model.h5')

# Test edilecek görüntüyü yükle
img = Image.open('test_photo.jpg').resize((224,224))
img_array = np.array(img)/255.0
img_array = np.expand_dims(img_array, axis=0)

# Tahmin yap
predictions = model.predict(img_array)
print("Tahmin:", np.argmax(predictions))
```

İpucu: Bu şekilde eğitilen model, kendi projelerinizde veya uygulamalarda gerçek zamanlı olarak çalışabilir.

7. Hata ve Öğrenme Deneyimi

- Model yanlış tahmin ederse, yeni veriler ekleyin.
- Eğitim ve test verilerini kullanarak modelin doğruluğunu artırın.
- Bu süreç, makine öğrenmesinin **“hata ile düzeltme ve özyinelemeli öğrenme”** mekanizmasını anlamanızı sağlar.

Örnek: Bir sınıf projesinde öğrenciler, farklı bitki yapraklarını tanıtan bir model kurabilir. Her yaprak türü için fotoğraflar ekler, modeli eğitir ve doğruluğunu test eder.

8. Kazanımlar

- Veri toplama ve etiketlemenin önemini öğrenmek
- Model eğitimi ve test sürecini deneyimlemek
- Hata ve doğruluk ilişkisini gözlemlemek
- Görüntü, ses ve poz verileri ile makine öğrenmesini pratikte görmek

Kendi Projeni Oluştur – Yapay Zekâyı Deneyimle!

Kendi yapay zekâ projenizi oluşturmak, öğrenmenin en eğlenceli ve etkili yoludur. Teachable Machine ile veri toplar, modeli eğitir ve tahminlerini test edersiniz. Her deneme, yapay zekânın nasıl öğrendiğini ve hatalardan nasıl ders çıkardığını doğrudan görmenizi sağlar. Kendi projelerinizde kararlar almak, veriyi seçmek ve modeli geliştirmek, teorik kavramları somut bir şekilde anlamanızı ve yaratıcılığınızı kullanmanızı sağlar. Yani bir model tasarlamak sadece eğlenceli değil, aynı zamanda yapay zekâyı *deneyimleyerek öğrenmenin anahtarıdır.*

TERİMLER SÖZLÜĞÜ

Algoritma:
Bir problemi çözmek veya bir görevi yerine getirmek için izlenen adım adım talimatlar dizisidir.

Aşırı Öğrenme (Overfitting):
Modelin eğitim verisini ezberlemesi ve yeni veriler üzerinde başarısız olması durumudur.

Büyük Veri (Big Data):
Çok büyük, hızlı ve farklı türlerde gelen veri kümeleridir. Geleneksel yöntemlerle işlenmesi zordur.

Bilgisayarlı Görü (Computer Vision):
Bilgisayarların görüntüleri ve videoları anlayabilmesini sağlayan yapay zekâ alanıdır.

ChatGPT:
İnsan dilini anlayıp metin üretebilen bir büyük dil modeli tabanlı yapay zekâ sistemidir.

Derin Öğrenme (Deep Learning):
Çok katmanlı yapay sinir ağları kullanarak karmaşık problemleri çözebilen makine öğrenmesi yöntemidir.

Denetimli Öğrenme (Supervised Learning):
Girdi ve doğru çıktılar (etiketler) kullanılarak modelin eğitildiği öğrenme yöntemidir.

Denetimsiz Öğrenme (Unsupervised Learning):
Verilerdeki gizli yapıları ve ilişkileri, etiket olmadan keşfetmeye çalışan öğrenme yöntemidir.

Doğal Dil İşleme (NLP):
Bilgisayarların insan dilini anlaması ve üretmesiyle ilgilenen yapay zekâ alanıdır.

Eğitim Verisi (Training Data):
Bir modelin öğrenmesi için kullanılan veri kümesidir.

Etiketleme (Labeling):
Verilere doğru cevapların veya kategorilerin eklenmesi işlemidir.

Generative AI (Üretken Yapay Zekâ):
Metin, görsel, müzik gibi yeni içerikler oluşturabilen yapay zekâ türüdür.

Gizli Katman (Hidden Layer):
Yapay sinir ağlarında giriş ve çıkış katmanları arasında bulunan ara katmanlardır.

Giriş Katmanı (Input Layer):
Verilerin modele ilk girdiği katmandır.

Hata (Error):
Modelin tahmini ile gerçek sonuç arasındaki farktır.

Halüsinasyon (AI Hallucination):
Yapay zekânın yanlış veya uydurma bilgi üretmesi durumudur.

Makine Öğrenmesi (Machine Learning):
Bilgisayarların verilerden öğrenmesini sağlayan yapay zekâ alt alanıdır.

Model:
Verilerden öğrenen ve tahmin yapan matematiksel yapıdır.

Nesne Tanıma (Object Detection):
Görüntülerdeki nesneleri tespit etme ve sınıflandırma işlemidir.

Otonom Sistemler:
İnsan müdahalesi olmadan kendi kendine karar verebilen sistemlerdir (örneğin sürücüsüz araçlar).

Önyargı (Bias):
Verilerdeki dengesizliklerden kaynaklanan ve modelin adaletsiz sonuçlar üretmesine yol açan durumdur.

Özyinelemeli Öğrenme (Iterative Learning):
Modelin kendini tekrar tekrar geliştirerek öğrenmesi sürecidir.

Pekiştirmeli Öğrenme (Reinforcement Learning):
Doğru davranışların ödüllendirildiği bir öğrenme yöntemidir.

Perceptron:
Yapay sinir ağlarının en temel yapı taşı olan basit yapay nörondur.

Prompt:
Yapay zekâya verilen komut veya girdidir.

Test Verisi (Test Data):
Modelin başarısını ölçmek için kullanılan veri kümesidir.

Veri:
İşlenmemiş bilgi parçalarıdır (sayılar, metinler, görüntüler vb.).

Veri Temizleme:
Hatalı, eksik veya gereksiz verilerin düzeltilmesi işlemidir.

Veri Gizliliği:
Kişisel verilerin korunması ve izinsiz kullanılmamasını ifade eder.

Yapay Sinir Ağı (Neural Network):
İnsan beyninden esinlenerek oluşturulan, birbirine bağlı yapay nöronlardan oluşan sistemdir.

Yapay Zekâ (Artificial Intelligence):
Bilgisayarların insan benzeri düşünme, öğrenme ve problem çözme yetenekleri kazanmasını sağlayan teknolojidir.

Çıkış Katmanı (Output Layer):
Modelin nihai sonucu ürettiği katmandır.